For Macintosh Programmers & Developers

# MacTech MAGAZINE

Get Ready for Mac OS 8!

Vol. 12, No. 8 • August 1996

## INSIDE:

**PROGRAMMING WORKSHOP**
Using Shared Libraries in CPX

**JAVATECH™**
Two Java Environments Compared

**FROM THE FACTORY FLOOR**
Heidi Ho!

**NEW TECHNOLOGIES**
Mac OS 8 Checklist

**OPENDOC**
Updating Parts Dynamically With SOM

**NEW TECHNOLOGIES**
Dragging and Rigid-Body Transformations

**PROGRAMMER'S CHALLENGE**
A-Maze-ing

**QUALITY CONTROL**
Debugging With QC

**GETTING STARTED**
Blinking Letters, Before

**SYMANTEC TOP 10**

**UNIFORM RESOURCE LOCATORS**

**AND MORE!**

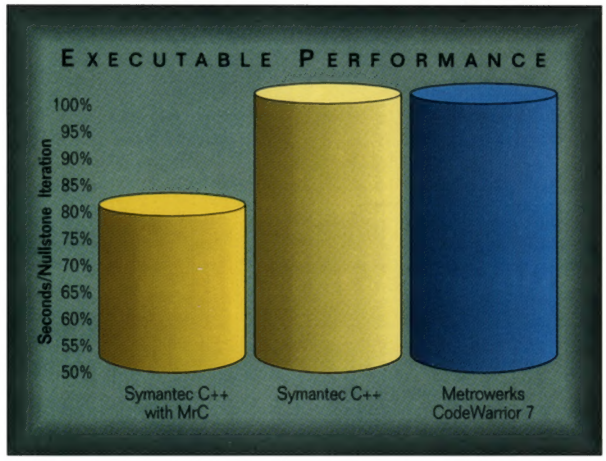
0 32128 74887 8 08

$5.85 US
$6.95 Canada
ISSN 1067-8360
Printed in U.S.A.

Now, through a joint development agreement, Symantec and Apple™ Computer let you produce the fastest Power Mac code.

Symantec C++ for Power Macintosh™ now comes with MrC– Apple's new optimizing compiler.

Industry-standard Nullstone tests show that




*Code compiled using MrC runs an average of 22% faster than code compiled with the standard Symantec C++ compiler or Metrowerks Code Warrior 7™.**

size and color. Organizing and navigating a project has never been easier.

In addition, Symantec C++'s multi-threaded environment gives you the ability to edit and write code while you compile. And our visual architect lets you quickly draw the interface. Corresponding code is then generated automatically.

# WHEN MRC GOES TO WORK, SYMANTEC C++ APPLICATIONS GET 22% FASTER.

applications compiled with MrC run an average of 22% faster.

## DEVELOP FASTER APPLICATIONS FASTER.

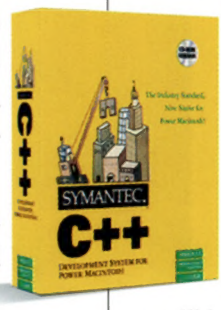
Not only can you develop the fastest Power Mac applications, you can write them fast, too. New AppleScript support lets you automate repetitive tasks. While the new linker provides fast turnaround for incremental builds.

With the fully integrated class browser, you'll quickly navigate your C++ class library. And support for templates and multiple inheritance further boosts your productivity.

## ORGANIZE AND MANAGE PROJECTS EASILY.

The new Project Manager lets you organize and manage nested projects. You can display hierarchical groups as folders within a project window for easier organization.

The editor gives you unlimited split panes and full text formatting– font,




MRC COMPILER
*produces the fastest Power Mac applications*

APPLESCRIPT
*automates the build process*

NEW INCREMENTAL LINKER
*provides fast incremental builds*

TEMPLATE AND MULTIPLE INHERITANCE SUPPORT
*increases productivity*

NESTED PROJECTS AND FOLDERS
*let you organize and navigate projects*

MULTI-THREADED ENVIRONMENT
*lets you edit and write code while compiling*

VISUAL ARCHITECT
*builds your interface visually*

All in all, Symantec C++ is a great way to develop the fastest applications for Power Mac.

## THREE CDS FOR THE PRICE OF ONE.

When you register as a Symantec C++ owner, you'll be enrolled in the Symantec C++ Subscription for Macintosh program. Subscribers will automatically receive two free product updates (on CD ROM) so you'll always have the latest features and tools.

*Learn more about Symantec C++ on the Internet at www.symantec.com Or call 1-800-628-4777, Extension 9H25 for more information.*

SYMANTEC.®

Offer valid in U.S.A. only. *Industry standard Nullstone Tests run 9/29/95 on Power Mac 6100/60. For more information in Canada, call 1-800-365-8541. In Australia, call 2-879-6577. In Europe, call 31-71-353111.
Symantec is a registered trademark of Symantec Corporation. All other trademarks are the property of their respective holders. All rights reserved. © 1995 Symantec Corporation.

# SmalltalkAgents®

DYNAMIC DEVELOPMENT TOOL

**Version 2.1**

Accelerated for Power Macintosh

## PowerPC Acceleration

SmalltalkAgents 2.1 breaks new ground in integrating 68K and PowerPC code for performance. STA 2.1 introduces a plug-in mechanism for extending and enhancing the virtual machine with fragment-based accelerators.

## Code Fragment Manager

STA2.1 supports the Code Fragment Manager (CFM). This feature provides direct access to native PowerPC code libraries. Not only can you more easily call code written in other languages, you can also take full advantage of the increasing variety of libraries available from Apple® and third parties.

## SmartComponents Library

The SmartComponents library contains new classes for building intelligent

user-interface components. These classes support greater-than-32,000 pixel coordinate systems, automatic scrolling, and improved drag-and-drop.

## Greater-than-32K Text

We have replaced TextEdit with the WorldScript® Aware Styled Text Engine (WASTE). WASTE not only removes the 32K barrier, but also provides WorldScript awareness, styled text, and embeddable objects such as graphics.

Q·K·S
Quasar Knowledge Systems, Inc.
1405 Main Street, PO Box 371478
Montara, CA 94037 USA
Tel: 415-728-5333, FAX: 415-728-1757

**Contact us today at <info@qks.com> to reserve your copy of SmalltalkAgents Version 2.1 or visit our web site at http://www.qks.com/.**

Mac™OS

For Macintosh Programmers & Developers

MacTech™ MAGAZINE

A publication of XPLAIN CORPORATION

# How To Communicate With Us

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail?** If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

| DEPARTMENTS | Internet | CompuServe | AppleLink |
|---|---|---|---|
| **Orders, Circulation, & Customer Service** | cust_service@mactech.com | 71333,1063 | – |
| **Press Releases** | press_releases@mactech.com | – | – |
| **Ad Sales** | ad_sales@mactech.com | – | – |
| **Editorial** | editorial@mactech.com | 71333,1065 | – |
| **Programmer's Challenge** | prog_challenge@mactech.com | – | – |
| **Online Support** | online@mactech.com | – | – |
| **Accounting** | accounting@mactech.com | – | – |
| **Marketing** | marketing@mactech.com | – | – |
| **General** | info@mactech.com | 71333,1064 | MacTechMag |
| **Online support area** | http://www.mactech.com | | |

## MACTECH MAGAZINE

*MacTech Magazine is grateful to the following individuals who contribute on a regular basis. We encourage others to share the technology.*

*We are dedicated to the distribution of useful programming information without regard to Apple's developer status.*

*For information on submitting articles, ask us for our **writer's kit** which includes the terms and conditions upon which we publish articles.*

### *Editors*

**Publisher** • Neil Ticktin
**Editor-in-Chief** • Will Iverson
**Editor** • Matt Neuburg
**Editor Emeritus** • Scott T Boyd, QKS
**Editor-at-Large** • Eric Gundrum

### *Contributing and Technical Editors*

**Copland** • Steve Kiene, Mindvision
**Internet** • Jon Wiederspan
**MagicCap/TeleScript** • Richard Clark, General Magic
**OpenDoc** • Tantek Çelik, 6prime corporation
**Performance Programming** • Jim Gochee, Connectix
**Tips & Tidbits** • Steve Sisak
**MacDev-1™** • Rich Siegel, Bare Bones Software, Inc.

### *Regular Columnists*

**Getting Started** • Dave Mark, Metrowerks
**Programmer's Challenge** • Bob Boonstra
**Insider's View** • Jordan J. Mattson, Apple Computer, Inc.
**Symantec Top 10** • Symantec Technical Support
**URLs** • Jim Straus, Key Internet Services

## XPLAIN CORPORATION

**Chief Executive Officer** • Neil Ticktin
**Chief Operating Officer** • Andrea J. Sniderman
**Controller** • Brian Shin
**Advertising Executive** • Ruth Subrin
**Senior Art Director** • Judith Chaplin
**Customer Service** • Al Estrada
**Vendor Relations** • Duk Im
**Administrative Assistant** • Susan Pomrantz
**Network Administrator** • Donal Corcoran
**Software Engineer** • Don Bresee

**Board of Advisors** • Steven Geller, Jordan Mattson, Blake Park, and Alan Carsrud

PRINTED WITH SOY INK™

This publication is printed on paper with recycled content.

All contents are Copyright © 1984-1996 by Xplain Corporation. All rights reserved. MacTech, MacTech Magazine, MacTech CD-ROM, MacTech Web, THINK Reference, Developer Depot, Sprocket, JavaTech, WebTech, MacDev-1, Developer Central, Virtual Developer Central and the MacTutorMan are trademarks of Xplain Corporation. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders.

**MacTech Magazine** (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are $47.00 per year. Canadian subscriptions are $59.00 per year. All other international subscriptions are $97.00 per year. Domestic source code disk subscriptions are $77 per year. All international disk subscriptions are $97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

**POSTMASTER**: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

# TABLE OF CONTENTS

August 1996 • Volume 12, No.8

For Macintosh Programmers & Developers

MacTech™ MAGAZINE

## FEATURE ARTICLES

**PROGRAMMING WORKSHOP** ..... 14
**Using Shared Libraries in CPX**
Can you really use the Code Fragment Manager and not go crazy?
— *By John Shackelford*

**JAVATECH™** ..... 29
**Two Java Environments Compared**
It just started brewing so it might be a little weak — *By Marty Wachter*

**NEW TECHNOLOGIES** ..... 36
**Mac OS 8 Checklist**
Ensuring your software will work properly — *By Steve Kiene*

**NEW TECHNOLOGIES** ..... 53
**Dragging and Rigid-Body Transformations**
QD GX can make your Mac a swinging place — *By Lawrence D'Oliveiro*

**QUALITY CONTROL** ..... 56
**Debugging With QC**
Stress your software, not yourself — *By Jeremy Vineyard*

**OPENDOC** ..... 73
**Updating Parts Dynamically With SOM**
Fast, automatic inter-part communication — *By Jeremy Roschelle*

## REGULAR COLUMNS

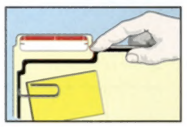
**GETTING STARTED** ..... 4
**Blinking Letters, Before** — *By Dave Mark*

**SYMANTEC TOP 10** ..... 33
— *By Rick Hartmann*

**PROGRAMMER'S CHALLENGE** ..... 60
**A-Maze-ing** — *By Bob Boonstra*

**UNIFORM RESOURCE LOCATORS** ..... 46
— *By Jim Straus*

**FROM THE FACTORY FLOOR** ..... 49
**Heidi Ho!** — *By Dave Mark*



**THE CLASSIFIEDS**
86

**NEWSBITS**
90

**TIPS & TIDBITS**
92

**ADVERTISER & PRODUCT INDEX**
94

*By Dave Mark*

# Blinking Letters, Before

I just got back from the World Wide Developer's Conference, where I got very little sleep but had a fantastic time. I saw Gil Amelio's long-awaited speech, reunited with friends I only get to see a few times a year, got a chance to hone my Java skills, smoked some cigars, and even took a small side trip to Napa Valley.

Before I got to the show, I wrote a small applet for this month's column. All puffed up with pride, I showed the applet to Peter Lewis (if you don't know Peter, check out Anarchie, along with his other cool Internet shareware, and peruse the May 1996 issue of *MacTech™ Magazine*, in which he features heavily), and asked him if he had any comments. What I got back from Peter was a complete rewrite that had a better design, and ran a lot faster to boot!

So here's what I'm going to do. This month, we'll go over my original applet, step by step. Once you understand the applet, take some time to think about how you might improve the design. Next month, we'll take a look at Peter Lewis' rewrite.

### The Original Applet

My applet started life as one of the standard sample Java applets that ships with the JDK. The original Blink was written by Arthur van Hoff (thanks, Arthur) and is shown in Figure 1. It paints a specified string in the applet window, then repeatedly and randomly redraws the words in the window in a series of related colors. (Some words are drawn in the background color, rendering them invisible.)

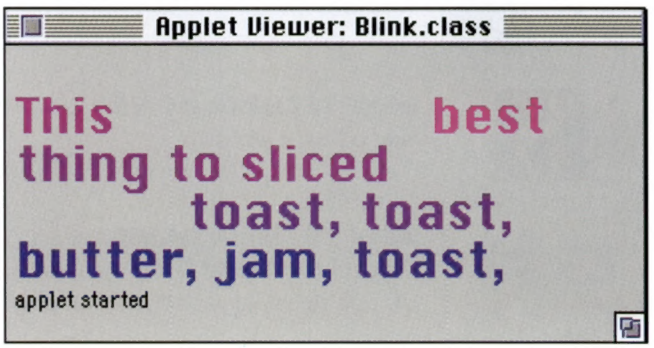



*Figure 1. The original Blink, as found in the Sun Java sample applets*

I rewrote Blink so that it would display its string in a bigger font, blinking the individual letters in completely random colors, as opposed to the related colors used by original Blink. Since I used CodeWarrior to build the applet, I renamed it CWBlink.

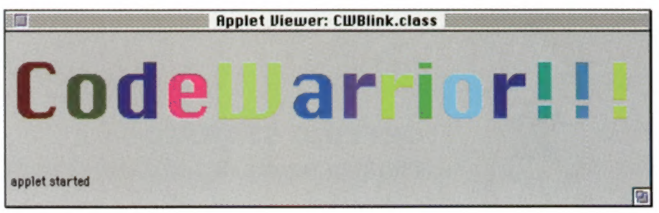
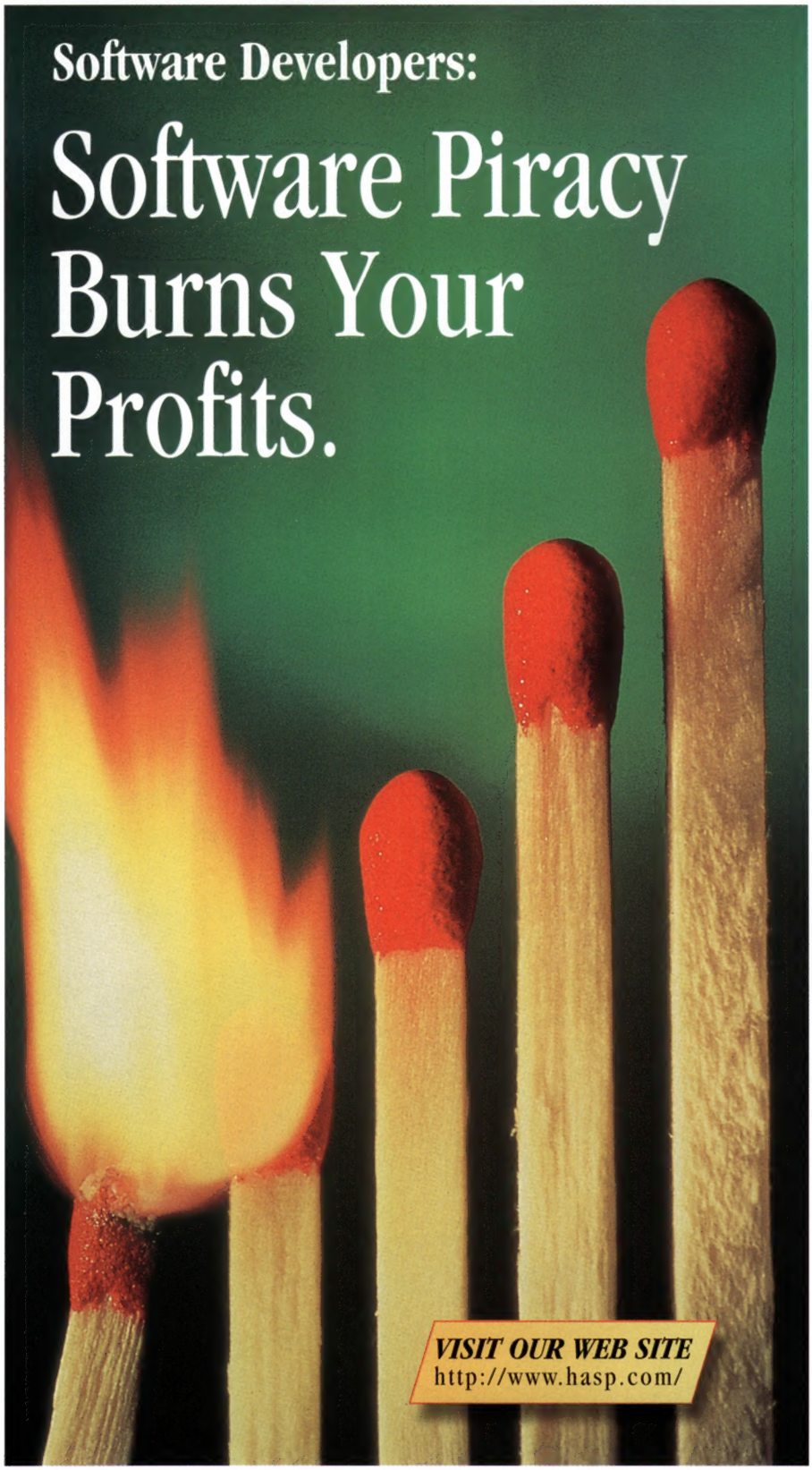



*Figure 2. My rewrite of Blink, now called CWBlink*

### Creating the CWBlink Applet

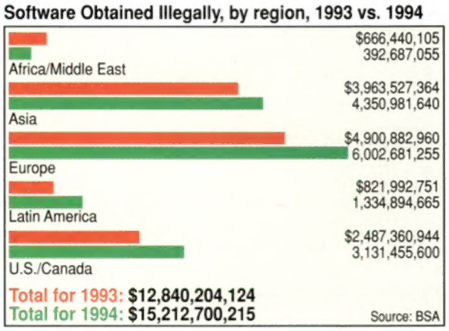
Here's how I created the CWBlink applet. I started in the CodeWarrior IDE. Since CodeWarrior uses a front-end/back-end architecture (front-end drop-ins for language, back-end drop-ins for target machines), they use the same IDE and debugger for Java as they use for C, C++, and Pascal.

Inside the IDE, I created a new project, using the Java Applet stationery (Figure 3).




Each year, the illegal use of software consumes nearly 50% of your potential revenues. With the flames of piracy eating away at your profits, can you afford not to protect your software?

**Software Obtained Illegally, by region, 1993 vs. 1994**




MacHASP® is widely acclaimed as the world's most advanced software protection solution for Macintosh computers. Since 1984, thousands of leading Mac and PC developers have used over one million MacHASP and HASP keys to protect billions of dollars worth of software. Why? Because MacHASP's security, reliability, and ease-of-use led them to a simple conclusion: MacHASP is the most effective software protection system available.

Today, more developers are choosing MacHASP than any other software protection method. To learn why, and to see how easily you can increase your revenues, call now to order your MacHASP Developer's Kit.

**1-800-223-4277**

**ALADDIN**

*The Professional's Choice*

| | |
|---|---|
| **North America** | **Aladdin Software Security Inc.**<br>Tel: (800) 223 4277, 212-564 5678<br>Fax: 212-564 3377<br>E-mail: sales@hasp.com<br>WWW: http://www.hasp.com/ |
| **Intl Office** | **Aladdin Knowledge Systems Ltd.**<br>Tel: 972-3-537 5795, Fax: 972-3-537 5796<br>E-mail: aladdin@aladdin.co.il |
| **United Kingdom** | **Aladdin Knowledge Systems UK Ltd.**<br>Tel: 01753-622266, Fax: 01753-622262 |
| **France** | **Aladdin France SA**<br>Tel: 1 40 85 98 85, Fax: 1 41 21 90 56 |

© Aladdin Knowledge Systems Ltd. 1985-1995 (6.95) HASP® is a registered trademark of Aladdin Knowledge Systems Ltd. All other product names are trademarks of their respective manufacrurers. Mac & the Mac OS logo are trademarks of Apple Computer, Inc., used under license

member of SPA

yes It runs with NetWare Developer tested only

Available and Ready for OS/2

MICROSOFT® WINDOWS™ COMPATIBLE

Mac™ OS

■ **Aladdin Benelux** 08894 19777 ■ **Aladdin Japan** 0426 60 7191 ■ **Aladdin Russia** 095 9230588 ■ **Australia** Conlab 3 8985685 ■ **Czech** Atlas 2 766085 ■ **Chile** Micrologica 2 222 1388 ■ **Denmark** Berendsen 39 577300 ■ **Egypt** Zeineldein 2 3604632 ■ **Finland** ID-Systems 0 870 3520 ■ **Germany** CSS 201 278804 ■ **Greece** Unibrain 1 6856320 ■ **India** Solutions 11 2218254 ■ **Italy** Partner Data 2 26147380 ■ **Korea** Dae-A 2 848 4481 ■ **Mexico** SiSoft 5 5439770 ■ **New Zealand** Training 4 5666014 ■ **Poland** Systherm 61 480273 ■ **Portugal** Futurmatica 1 4116269 ■ **Romania** Interactiv 64 153112 ■ **South Africa** D Le Roux 11 886 4704 ■ **Spain** PC Hardware 3 4493193 ■ **Switzerland** Opag 61 7169222 ■ **Taiwan** Teco 2 555 9676 ■ **Turkey** Mikrobeta 312 467 7504

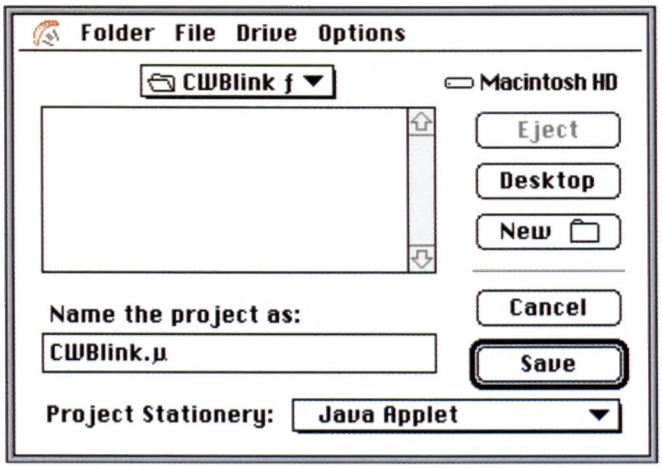



*Figure 3. The New Project dialog box, showing the Java Applet stationery*

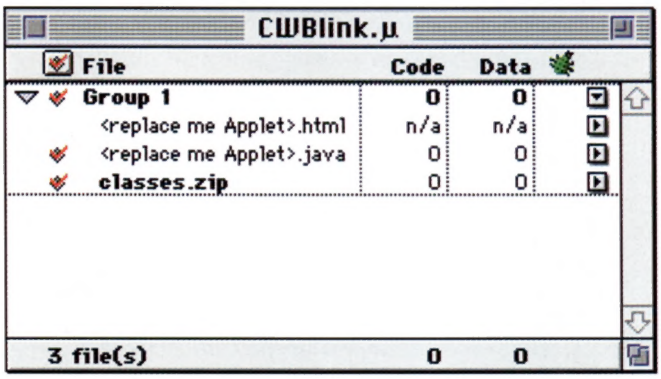



*Figure 4. The new Java Applet project window, before we replace the .html and .java files*

Once the project was created (Figure 4), the next step was to replace the .html and .java files with our own. First, I double-clicked the file <replace me Applet>.html in the Project window. When the stationery file opens (notice that it is marked as read-only), I selected **Save As...** from the **File** menu and saved the file as CWBlink.html. Next, I replaced the text in the .html file with the following:

```
<title>Blinking CodeWarrior</title>
<hr>
<applet codebase="CWBlink Classes" code="CWBlink.class"
width=530 height=100>
<param name=blinker value="CodeWarrior!!!">
<param name=speed value=1000>
</applet>
<hr>
<a href="CWBlink.java">The source.</a>
```

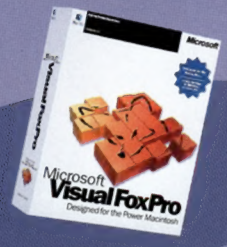
Once your HTML is typed in, save the code and close the file.

The HTML code should look familiar. Note that the `codebase` attribute tells the Java virtual machine the name of the folder containing your class files. The `code` attribute tells the VM the name of the class file to start with. The `blinker` parameter is the string that does the blinking. The `speed` parameter lets you change the speed of the blinking.

Next, double-click the file <replace me Applet>.java. Save the file as CWBlink.java. Replace the contents of the file with the following:

```
import java.awt.*;

public class CWBlink extends java.applet.Applet
    implements Runnable
{
  Thread blinkThread;
  String blinkString;
  Font     font;
  int      speed;

  public void init()
  {
    font = new java.awt.Font(
        "TimesRoman", Font.PLAIN, 64);

    String att = getParameter("speed");
    speed = (att == null) ? 400 :
        (1000 / Integer.valueOf(att).intValue());

    att = getParameter("blinker");
    blinkString = (att == null) ? "CodeWarrior!!!" : att;
  }

  public void paint(Graphics g)
  {
    int x = 0, y = font.getSize()-10;

    g.setColor(Color.black);
    g.setFont(font);
    FontMetrics fm = g.getFontMetrics();

    for (int index=0; index<blinkString.length(); index++ )
    {
      int red = (int)(Math.random() * 256);
      int green = (int)(Math.random() * 256);
      int blue = (int)(Math.random() * 256);

      char character = blinkString.charAt(index);
      int w = fm.charWidth(character);

      g.setColor( new java.awt.Color( red, green, blue ) );

      Character c = new Character( character );
      g.drawString(c.toString(), x, y );
      x += w;
    }
  }

  public void start()
  {
    blinkThread = new Thread(this);
    blinkThread.start();
  }

  public void stop()
  {
    blinkThread.stop();
  }

  public void run()
  {
    while (true)
    {
      try {Thread.currentThread().sleep(speed);}
      catch (InterruptedException e){}
    repaint();
    }
  }
}
```

**Ever since Fred got the new Visual FoxPro for Power Macintosh, he catches the Frisbee every time.**




And perhaps more importantly, he also gets his work done in record time. Thanks to Microsoft® Visual FoxPro™ 3.0 for Power Macintosh,® one might say that Fred's productivity is positively soaring.

After all, the powerful object technology of Visual FoxPro lets him create class libraries with inheritance and encapsulation and therefore reuse code quickly and easily. Fred has no trouble integrating data, either, whether it's located in applications, client/server databases, or local datastores. And because of Visual FoxPro's complete binary compatibility, he can develop in the familiar Mac™ environment and rest assured that all code, including Visual FoxPro objects, will run in Windows® 3.1 and Windows 95 operating systems with no modifications.

To find out more about the power of the Visual FoxPro 3.0 database management system, just point your mouse to http://www.microsoft.com/devonly/. Or give us a call at **(800) 621-7930, Dept. A334DS**, in the 50 United States.*

**Microsoft**

WHERE DO YOU WANT TO GO TODAY?™

*In Canada call (800) 563-9048; outside the U.S. contact your local subsidiary. ©1996 Microsoft Corporation. All rights reserved. Microsoft and Windows are registered trademarks and Visual FoxPro and *Where do you want to go today?* are trademarks of Microsoft Corporation. Power Macintosh is a registered trademark and Mac is a trademark of Apple Computer, Inc. Frisbee® is a trademark of Mattel, Inc. Used with permission.

The last step in setting up your project is to tell CodeWarrior how you want your applet classes saved. Select **Preferences** from the **Edit** menu. When the Preferences dialog appears (Figure 5), select the Java Project item under the Project heading. Make sure "Class Folder" is selected from the Project Type popup. Now type **CWBlink Classes** in the Folder Name field. This tells CodeWarrior to save all of your classes as .class files in a folder named CWBlink Classes, which is what's expected by the HTML you typed in earlier.

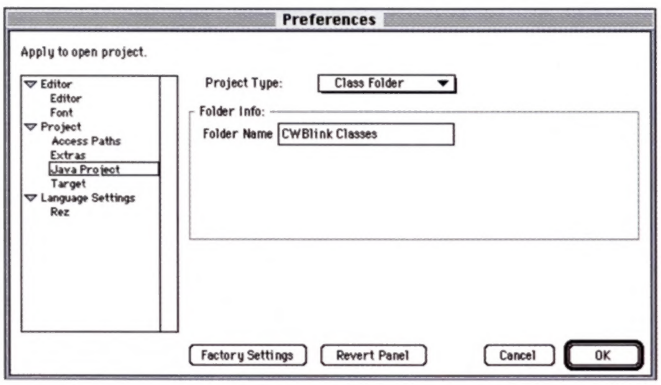



*Figure 5. The Java Project pane from CodeWarrior's Preferences dialog*

Select **Make** from the **Project** menu to compile your .java file and save out your new .class file. Once your code compiles, CodeWarrior will create a new folder called CWBlink Classes, then create a new class file named CWBlink.class inside the CWBlink Classes folder. Once that happens, drag your HTML file onto the Metrowerks Java application. Metrowerks Java will parse your HTML code and run your brand new blinking applet.

Let's take a look at the source code.

### CWBlink Source Code

CWBlink.java starts by importing the `java.awt` classes. Next, the `CWBlink` class is defined. Notice that `CWBlink` extends the `Applet` class (you've seen this before) and also implements the `Runnable` interface (something new). Here's what the Sun HTML doc has to say about the `java.lang.Runnable` interface:

> "This interface is designed to provide a common protocol for Objects that wish to execute code while they are active. For example, Runnable is implemented by class Thread. Being active simply means that a thread has been started and has not yet been stopped.
>
> "In addition, Runnable provides the means for a class to be active while not subclassing Thread. A class that implements Runnable can run without subclassing Thread by instantiating a Thread instance and passing itself in as the target. In most cases, the Runnable interface should be used if you are only planning to override the run() method and no other Thread methods. This is important because classes should not be subclassed unless the programmer intends on modifying or enhancing the fundamental behavior of the class."

Basically, extending an existing class is the traditional subclassing mechanism you are used to from C++. An interface defines a set of methods and variables that will be available from within any class implementing that interface. For example, to implement the `Runnable` interface, you must provide a method named `run()` that gets executed when your object is activated. As the HTML summary indicates, the `Runnable` interface is implemented by the `Thread` class. The `run()` method is where the thread's action takes place.

```
import java.awt.*;

public class CWBlink extends java.applet.Applet
    implements Runnable
{
  Thread blinkThread;
  String blinkString;
  Font     font;
  int      speed;
```

The `init()` method is called when the applet is created. We'll start off the initialization process by creating a `Font` object that defines how the blinking text will appear. Next, we'll load the `speed` and `blinker` parameters. Note that `speed` gets a default value of 400 (otherwise, the value is 1000 divided by the provided value) and `blinkString` gets a default value of `"CodeWarrior!!!"`.

```
  public void init()
  {
    font = new java.awt.Font(
        "TimesRoman", Font.PLAIN, 64);

    String att = getParameter("speed");
    speed = (att == null) ? 400 :
        (1000 / Integer.valueOf(att).intValue());

    att = getParameter("blinker");
    blinkString = (att == null) ? "CodeWarrior!!!" : att;
  }
```

`paint()` gets called when the applet's pane gets an update event. We'll draw the `blinkString` at the `x` and `y` coordinates. We use the font's point size to calculate the point where we draw the string in the applet's pane.

```
  public void paint(Graphics g)
  {
    int x = 0, y = font.getSize()-10;
```

We'll set the foreground color to black and the current font to `font`. We'll also build a `FontMetrics` object from the current `Graphics` environment. To learn about `FontMetrics`, see the HTML file java.awt.FontMetrics. This will give us access to methods like `charWidth()` which lets

Announcing

# MICROGUARD PLUS.™

# Why so many developers are switching to MicroGuard copy protection

## MicroGuard is committed to uncompromising technological superiority

"Technology at its peak" is our commitment to you. That is why we have brought you **MicroGuard Plus**™. And, MicroGuard Plus is **100% backwards** compatible with MicroGuard. This means MicroGuard and MicroGuard Plus can be used interchangeably. Just as we promised!

## MicroGuard has now surpassed its own technological lead, actually improving on the best

MicroGuard Plus is everything MicroGuard is, plus 40-bit encryption, two additional passwords, 64-Bit Array, 32-byte public area, 45% size reduction, enhanced counter and more. In addition, MicroGuard Plus offers two new utilities: **QuickGuard**™ and EasyGuard™ allow you to protect your applications without touching your source code. The only feature that is not plus is the price. :-)

## MicroGuard offers you the most sophisticated network protection

Our network protection, **MicroGuard Net**™, is so superior, we had to hire an Apple network engineer to execute our specifications.

## MicroGuard is the best selling Macintosh key in the world

MicroGuard sells more Macintosh copy-protection keys than anyone else in the world!

## MicroGuard is the only key developed by Mac developers, and is the first and only 100% ADB savvy key

We have been developing Mac applications as a seed development house since 1984. We are not a PC protection company that has come to you with a Mac product. MicroGuard is fully ADB savvy and offers extended addressing. Unlike other protection devices, MicroGuard never clashes with other keys. Only MicroGuard offers this level of sophistication.

## MicroGuard delivers developer support within 24 hours

We will answer any inquiry you have within 24-hours. We also have a fully loaded AppleLink bulletin board which contains all our libraries, tech notes, Q&A and nearly everything you'll ever need!

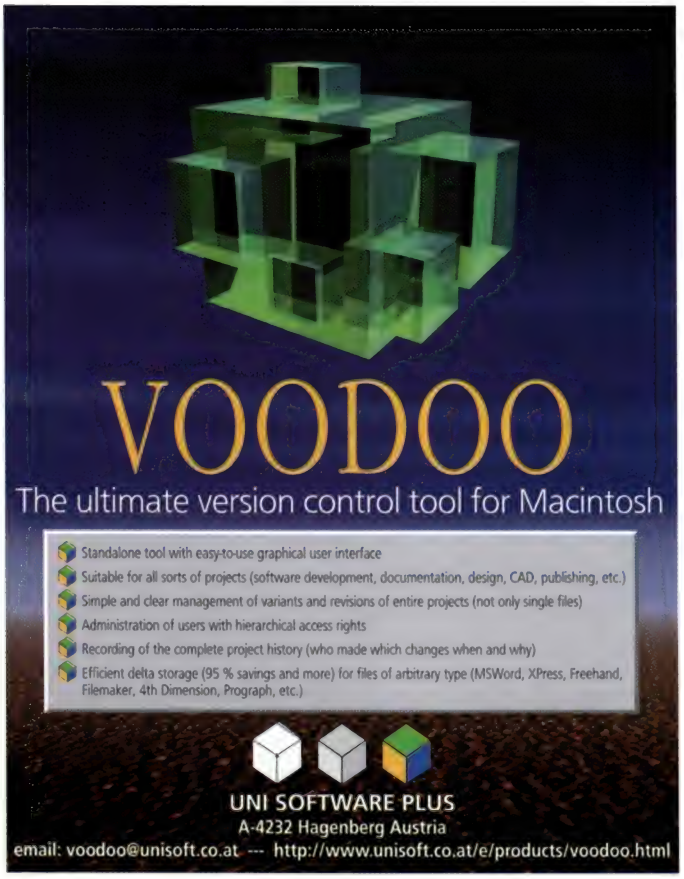
(actual size)

*For more information and to order a Developer's Kit or to receive a free CD ROM about MicroGuard, please contact us at:*

MicroGuard USA: Tel: (303) 320-1628 • Fax: (303) 320-1599 • AppleLink: M.GUARD
International: Tel: (972) 3 558-2345 • Fax: (972) 3 558-2344 • AppleLink: MICROGUARD

Mac™OS

Pirate Busters


VOODOO
The ultimate version control tool for Macintosh
Standalone tool with easy-to-use graphical user interface
Suitable for all sorts of projects (software development, documentation, design, CAD, publishing, etc.)
Simple and clear management of variants and revisions of entire projects (not only single files)
Administration of users with hierarchical access rights
Recording of the complete project history (who made which changes when and why)
Efficient delta storage (95 % savings and more) for files of arbitrary type (MSWord, XPress, Freehand, Filemaker, 4th Dimension, Prograph, etc.)
UNI SOFTWARE PLUS
A-4232 Hagenberg Austria
email: voodoo@unisoft.co.at --- http://www.unisoft.co.at/e/products/voodoo.html

us calculate the proper x-coordinate for the next letter to be drawn, based on the width of the current character.

```
g.setColor(Color.black);
g.setFont(font);
FontMetrics fm = g.getFontMetrics();
```

Next, we'll step through the `blinkString`, one letter at a time. For each letter, we'll calculate a random value for `red`, `green`, and `blue`. We'll store the current character in `character`, then calculate the character's width.

```
for (int index=0; index<blinkString.length(); index++ )
{
  int red = (int)(Math.random() * 256);
  int green = (int)(Math.random() * 256);
  int blue = (int)(Math.random() * 256);

  char character = blinkString.charAt(index);
  int w = fm.charWidth(character);
```

We'll set the current color using our random `red`, `green`, and `blue` values. Next, we'll create a new `Character` object (note that `Character` is different from `char` – see `java.lang.Character`), which gives us an easy way to convert a single character to a string. We need a string to pass to `drawString()`, and the `Character` wrapper gives us an easy way to get there. If anyone out there can think of a simpler way to draw the character on the screen at the specified x and y location, please let me know. Just be sure you test your theory before you write!

Once we draw the character, we'll bump x by the width of the character.

```
    g.setColor( new java.awt.Color( red, green, blue ) );

    Character c = new Character( character );
    g.drawString(c.toString(), x, y );
    x += w;
  }
}
```

`start()` is similar to `init()`, but gets called when the applet's document is visited. We'll create a new `Thread` object and call its `start()` method. When the applet's document is no longer on the screen, the `stop()` method is called. In our case, we'll call the `Thread`'s `stop()` method to stop the thread.

```
public void start()
{
  blinkThread = new Thread(this);
  blinkThread.start();
}

public void stop()
{
  blinkThread.stop();
}
```

Remember that the `run()` method contains the body of the thread we are starting up. Our thread enters an infinite loop. The loop tries a `sleep()` for `speed` milliseconds. We catch the `InterruptedException` exception (which occurs when another `Thread` interrupts this thread), but we do nothing about it. The loop ends by forcing a repaint of the applet.

```
public void run()
{
  while (true)
  {
    try {Thread.currentThread().sleep(speed);}
    catch (InterruptedException e){}
  repaint();
  }
}
```

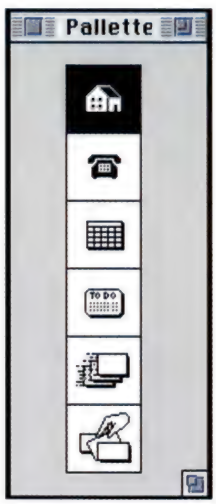
### Till Next Month...

This month's applet introduced the concept of `FontMetrics`, type wrappers (in our case, `java.lang.Character`), and `Threads`, along with a quick tour of CodeWarrior's Java environment. Next month, we'll take a look at Peter Lewis' rewrite of this applet. Think about how you might improve the design of this applet. What changes would you make? Peter's changes were very interesting and definitely worth exploring. See you next month.

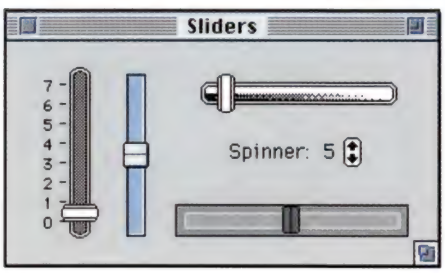
# AppMaker

## Your Assistant Programmer

AppMaker makes it fast and easy to build a user interface. Just point and click, then let AppMaker create resources and generate source code. AppMaker generates excellent source code for most popular languages and frameworks.



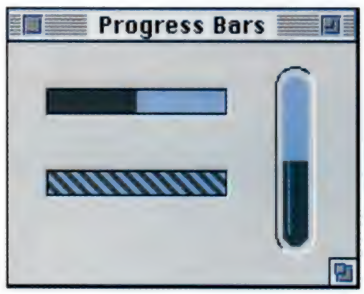

It's like having an expert assistant working for you. Use it as a prototyping and productivity tool. You concentrate on the unique features of your application instead of the standard parts. Novices read the AppMaker code to learn Macintosh programming techniques.



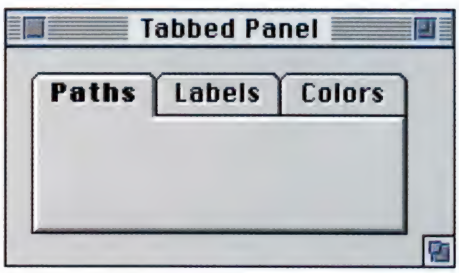

AppMaker now supports a wide variety of gadgets including 3D controls, picture buttons, sliders, progress bars, and tabbed panels. Users can easily customize and extend AppMaker to support other custom CDEFs.

The built-in simulator lets you try out all these new gadgets.

Interested in OpenDoc and the OpenDoc Development Framework (ODF)? AppMaker makes it easy to get started. Just design your interface then generate code to make an OpenDoc part editor.




AppMaker also supports CodeWarrior PowerPlant, procedural C, and PowerMac development. A single AppMaker document can now generate code and resources for a wide variety of languages, frameworks, and eventually, platforms.




Includes one-year automatic update service with at least three CDs per year.

Available from MacTech Mail Order Store for just $299 with a 60-day money-back guarantee.

Demo versions on AOL: keyword Bowers, on Compuserve in the MacDev forum, and at ftp://members.aol.com/bowersdev.

B • O • W • E • R • S
Development

P.O. Box 929, Grantham, NH 03753 • 603-863-0945
bowersdev@aol.com • 70731.3710@compuserve.com

# Our competitors' habit of charging for each piece of their installer program is a bit puzzling.

- CREATE UP TO 128 EXPERT-INSTALL "PACKAGES"
- IMPROVED DEVELOPER CUSTOMIZATION

The reason we call StuffIt InstallerMaker 3.0 the Complete Installation Solution is because we include **everything you need** to prepare a professional package of files for installation on your user's machines. An installer, updater, and an uninstaller are just three components included for the same price one of our fine competitors charges for just the installer. Using our installer reduces technical support calls due to end-user errors during installation, saves disks (if distributing on floppies) or download time (if distributing on-line). You will save more than the cost of the Installer license by significantly reducing distribution costs. And the puzzle will be solved.

You'll have every piece in place.

Download a **free,** fully-functional copy of StuffIt InstallerMaker 3.0 from www.aladdinsys.com, or call (408) 761-6200 and ask for Developer Sales.

**STUFFIT INSTALLERMAKER 3.0**

© 1996 Aladdin Systems, Inc. 165 Westridge Drive, Watsonville, CA 95076. Fax: (408) 761-6206. Internet: cust.service@aladdinsys.com. AOL, AppleLink: ALADDIN. StuffIt InstallerMaker is a trademark of Aladdin Systems, Inc. Other products are trademarks of their respective holders.

• Built-in Resource Compression
• Uninstall
• Custom Destinations
• Built-In Updaters
• Best Compression Available
• Full Scripting and Recording
• Move/Copy/ Rename any File
• Resource Installation
Aladdin Systems

*By John Shackelford*

# Using Shared Libraries in CPX

***Can you really use the Code Fragment Manager and not go crazy?***

## Introduction

The Power Mac and upgrades to System 7 harbor many exciting new technologies waiting for application developers to explore. One new technology allows executable code to be shared at run-time. This means you can share code libraries across applications regardless of which development environment you use – or in other words you can finally start using all that code those C++ developers are writing down the hall from you! This technology is provided by a new manager called the Code Fragment Manager (CFM). The CFM is responsible for loading fragments, blocks of executable PowerPC code and data, into memory, and preparing them for execution. Currently the CFM is supported only on Power Macs. However, very soon the CFM will be supported on 68K Macs.

This article describes and demonstrates with some example code one approach to accessing shared libraries from Prograph CPX. It shows a way to wrap a C/C++ library, defining an XPrim that provides the interface to shared libraries and describing in detail a set of CPX classes which encapsulate the use of shared libraries. You'll receive many benefits from using shared libraries in your applications, including reduced development time (because non-Prograph code no longer has to be ported to the Prograph language) and increased application speed (because of the native Power Mac code). The code described in this article gives you the basic tools for accessing shared libraries, listing export symbols, connecting to shared libraries, and executing functions. The code is shareware, and can be found at Tangent Systems' Web page.

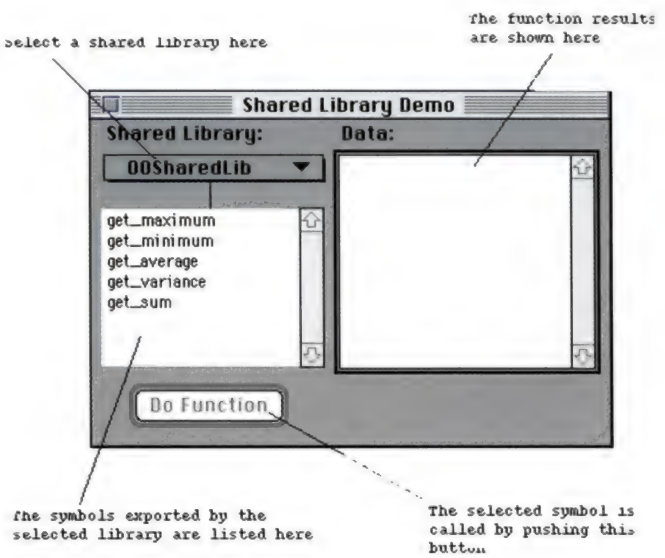



*Figure 1. The main window for the demo app*

## Shared Library Demo Application

For this article I have created a simple demo application and two shared libraries. The demo application will access the

***John Shackelford*** is a founding partner, and Director of Product Development, of Tangent Systems, a software development company based in San Diego. When he's not playing with his three children, he's busy at work writing CPX code. He can be reached at shackx@aol.com. Tangent Systems' Web page is at http://www.tangentsys.com.

***Eddy Award Winner** for **Best New Developer Tool***
*– MacUser Editors Choice Awards, 1993*

*"A distinct improvement over ResEdit."*
*– MacTech / MacTutor*

*"Resorcerer's data template system is amazing!"*
*– Bill Goodman, author of Compact Pro*

*"Nuke ResEdit! Resorcerer is mission-critical for us."*
*– Dave Winer, Userland Frontier*

*"The color pixel editors are wonderful! A work of art!"*
*– Dave Winzler, author of Microseeds Redux*

*"Every Macintosh developer should own a copy of Resorcerer."*
*– Leonard Rosenthol, Aladdin Systems*

*"Resorcerer will pay for itself many times over in saved time and effort."*
*– MacUser review*

*"The template that disassembles 'PICT's is awesome!"*
*– Bill Steinberg, author of Pyro! and PBTools*

*"Resorcerer proved indispensible in its own creation!"*
*– Doug McKenna, author of Resorcerer*

*"...a wealth of time-saving tools."*
***MacUser Review, Dec. 1992***

# RESORCERER®

## Version 1.2.4

## The Resource Editor for the Macintosh Wizard

### ORDERING INFO

Needs: ≥Mac Plus, ≥ Sys 4.2, 1MB
Likes: ≥Mac Plus, ≥ Sys 7.0, 2MB
32-bit clean, AU/X compatible

Price: $256 (decimal)
(Educational, quantity, or other discounts available)

Includes: 500 page manual
60-day Money-Back Guarantee
Domestic UPS ground shipping

Payment: Check, PO's, or Visa/MC

Extras (call us):
COD, FedEx, UPS Blue/Red, International Shipping

**Downloadable Demos/Updaters:**
AppleLink: Software Sampler
AOL: Software Libs/Development
CompuServe: MACDEV/Tools
or call us.

**New 1.2 Features:**

- **New 'cicn', 'ppat', 'crsr', 'acur', 'pltt', 'clut' editors**
- **Powerful icon family editing (all 9 icon types)**
- **Color pixel anti-aliasing, dithering, and lots more**
- **Complete 'PICT' disassembly and reassembly**
- **Resource sorting; ROM resource browsing**
- **120 template field parsing types now supported**
- **New insertion & deletion template field types**
- **Text-only 'PICT' resources**
- **Lots of improvements throughout**

- Easier, faster, more Mac-like, and more productive than ResEdit
- Safer memory-based, not disk-file-based, design and operation
- All file information and common commands in one easy-to-use window
- Compares resource files, and even **edits your data forks** as well
- Visible, accumulating, editable scrap
- Searches and opens/marks/selects resources by text content
- Makes global resource ID or type changes easily and safely
- Builds resource files from simple Rez-like scripts
- Most editors DeRez directly to the clipboard
- All graphic editors support screen-copying or partial screen-copying
- Hot-linking Value Converter for editing 32 bits in a dozen formats
- Its own 32-bit List Mgr can open and edit very large data structures
- Templates can pre- and post-process any arbitrary data structure
- Includes nearly 200 templates for common system resources
- TMPLs for Installer, MacApp, QT, Help, AppleEvent, OCE, GX, etc.
- Full integrated support for editing color dialogs and menus
- Try out balloons, 'ictb's, lists and popups, even create C source code
- Integrated single-window Hex/Code Editor, with patching, searching
- Editors for cursors, versions, pictures, bundles, and lots more
- Well-designed, helpful developer tools being added all the time
- Relied on by thousands of Macintosh developers around the world

MATHEMÆSTHETICS, INC.
P.O. Box 298 • Boulder • CO • 80306-0298 • USA
Phone: (303) 440-0707 • Fax: (303) 440-0504
AppleLink/AmericaOnline: RESORCERER • Internet: resorcerer@aol.com

shared libraries. The libraries each contain a different implementation of the same set of functions. One library (OOSharedLib) implements a set of functions using C++ code, while the other library (ExampleCLib) uses all C code.

When the application launches and opens its main window (Figure 1), it also "opens" or "connects to" the two shared libraries. The exported symbols or functions contained in each library are then available for the application to use. The functions or symbols that a library exports are listed in the main window. The functions in each library have been designed to operate on an array of data, and can compute the maximum, minimum, average, variance, or sum of the data.

You can select the desired library by choosing its name in the pop-up menu. The scroll list on the left side of the window lists the exported symbols of the selected shared library. When a symbol is selected in this scrolling list, the **Do Function** button becomes active. When the **Do Function** button is pushed, the selected function in the selected library is executed. The results are then listed in the right-hand scroll list.

The pop-up menu in the window lists the two shared libraries the application will work with. These libraries are loaded when the main window opens by a call to the window's method `/loadLibraries` (Figure 2). A `«Shared Library»` instance is created for each shared library and stored in an attribute of the main window for later use.

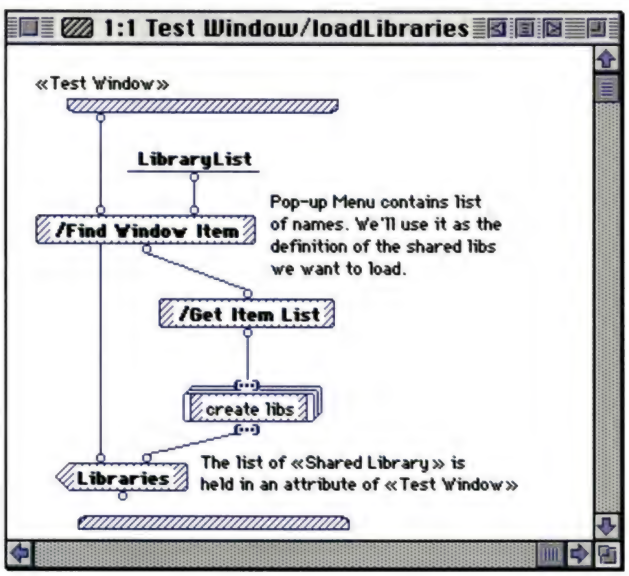

1:1 Test Window/loadLibraries
«Test Window»
LibraryList
/Find Window Item
Pop-up Menu contains list of names. We'll use it as the definition of the shared libs we want to load.
/Get Item List
create libs
Libraries
The list of «Shared Library» is held in an attribute of «Test Window»

*Figure 2. Test Window has a method which loads a list of shared libraries*

Exported symbols are the items which can be referenced by other code fragments. You can think of exported symbols as the public functions the library supports. Internally there may be a lot of functions that are hidden from the outside world. This provides some measure of control over how the library is used – which can be a nice feature for developers and users. As a developer, I may not always want every function in the library to be exposed, for fear they might not be properly used; and as a user, I may not want to know all the gory details of someone else's library. The typical Power Mac development environment has a procedure you must follow to define the export symbols. In CodeWarrior, for instance, there is a file (`*.exp`) you can edit prior to building the shared library, that controls which symbols will be exported (more on this later on).

Now let's try one function with the demo; then we will dive into more Prograph code. With the application still running, select a function in the function list and click **Do Function** (or simply double-click a function in the function list).

The demo application on the CPX side of things generates a list of 50 numbers. It then sends this list (embedded in a data structure) as an argument to the chosen function in the shared library. The shared library function then calculates the result and sends it back to the application (Figure 3). For the case where we use OOSharedLib and its `get_average` function, I did some timing analysis and found that a pure Prograph-coded equivalent using CPX built-in list primitives can be slower (I compared 68K compiled CPX to using a PPC-native shared library). Using 8000 values on a Power Mac 7100/80 (System 7.5), I got these results:

Interpreted CPX . . . . . 200 Ticks
Compiled CPX . . . . . 20 Ticks
Using native shared library . . < 1 Tick

(By the way, in looking at my Prograph code, don't be put off by the idiosyncracies of my shorthand comments to myself, where I'm reminding myself what data type is being input or output. "`<Shared Library>`" means `«Shared Library»`, an instance of the `Shared Library` class; at the time I was writing that comment I didn't know how to type the '«' character! Use of parentheses generally indicates a list; so, for instance, "`():name`" means "a list of names".)

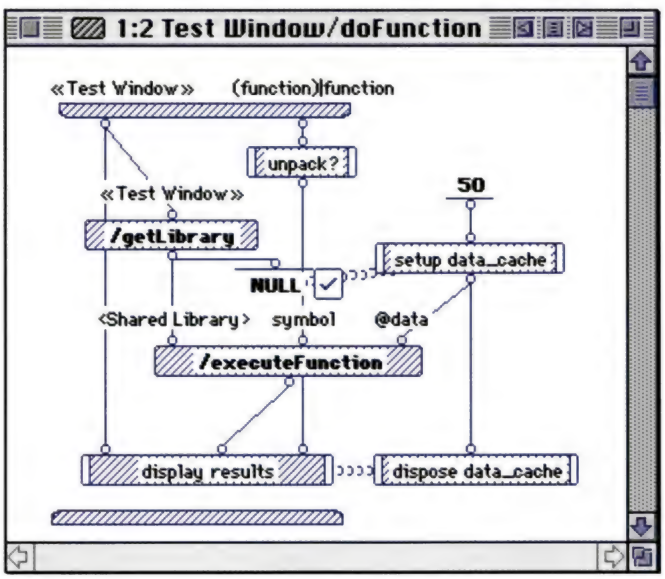



*Figure 3. The method /doFunction handles the execution of the selected library function*

# Power Too Abundant to Meter

neoAccess™
Cross-Platform Object Database Engine

## Powerful

NeoAccess and NeoShare are the most powerful object-oriented database engines available. They display electrifying performance—up to ten times that of competitors. Behind an elegant programming interface is a high performance query engine utilizing: extended binary trees and binary search algorithms tuned for short access times, dynamically combined, collapsed, and compressed indices, and object caching for lightning fast access to previously used objects.

## No Runtime Fees

Get the power of NeoAccess and avoid the expense and administrative hassle of feeding the runtime fees meter. You pay one affordable price no matter how many copies of your application you sell or use.

## Cross Platform

Others may promise cross-platform development tools—NeoLogic delivers. NeoAccess and NeoShare consists of C++ classes designed for use with popular compilers and application frameworks on Windows® (3.x, '95 and NT), Macintosh® (M68k and PPC), and Unix™ (all flavors) platforms. Full source code is available so both products can even be used with custom frameworks.

## New in NeoShare Version 2.0

Cross-platform and internet-savvy NeoShare 2.0 is now shipping! NeoShare extends standard application frameworks including; Metrowerks' PowerPlant, Symantec's THINK Class Library and Apple's MacApp on the Macintosh, and Microsoft's Foundation Classes and Borland's OWL in Intel-based environments.

neoShare™
Client/Server Object Database Engine

## Scalable

NeoShare extends the core features of NeoAccess to client/server applications. NeoShare includes the NeoAccess Toolkit and everything you need to create data intensive distributed applications. Its advanced client/server architecture provides shared access to objects by multiple inter- and intra-networked clients. Combine client and server functions into a single application or create separate client and server applications. With NeoShare you have complete flexibility in the design of your distributed systems.

## Proven

Thousands of commercial and in-house developers have already found that NeoLogic's technology enabled them to build fast, lightweight and powerful applications in record time. That's why NeoAccess and NeoShare based applications are already operating on millions of computers. Tap into the power for your next development project!

neo•logic™

Powering Development of Object-Oriented Applications

**NeoLogic Systems** 1450 Fourth St., Suite 12 v. 510.524.5897
neologic@neologic.com Berkeley, CA 94710 f. 510.524.4501

The local `setup data_cache` builds up a vector of data in a data structure to be sent off to the shared library. Figure 4 shows the fields of this structure. The functions in the shared library have been designed to return the result in the `result` field.

**External Structure:** data_cache

**Size:**20

| Fields | Type |
|---|---|
| length | unsigned long |
| success | Int4 |
| result | Real8 |
| data | Real8@ |

*Figure 4. The data_cache fields as shown in the CPX "info" window*

### SLD Kit CPX interface

On the CPX side of the things, the interface for using shared libraries is handled by a class called `Shared Library`. For Power Mac–native shared libraries you'll actually use its subclass `Shared Library PPC`. When the port to 68K machines is complete, CPX users will use the `Shared Library 68K` subclass. The purpose of the `Shared Library` class is to encapsulate all the low-level calls to the CFM and the Mixed Mode Manager when working with shared libraries. The class provides a set of methods to access shared libraries. We will now review the six most significant of these methods.

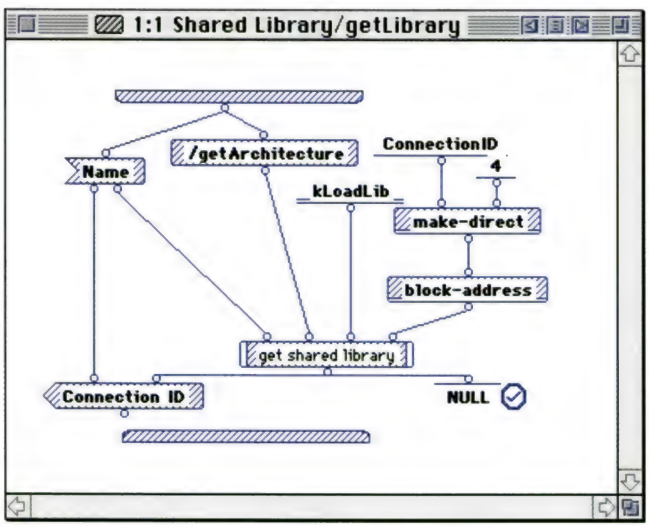



*Figure 5. /getLibrary gets a Connection ID to the shared library*

### /getLibrary

This method (Figure 5) retrieves a connection identification number using the toolbox call `GetSharedLibrary`. That call is in the local `get shared library`. The attribute `Connection ID` comes into play when using functions in the shared library or listing the exported symbols.

### /closeLibrary

This method uses the `Connection ID` to close the connection (Figure 6).

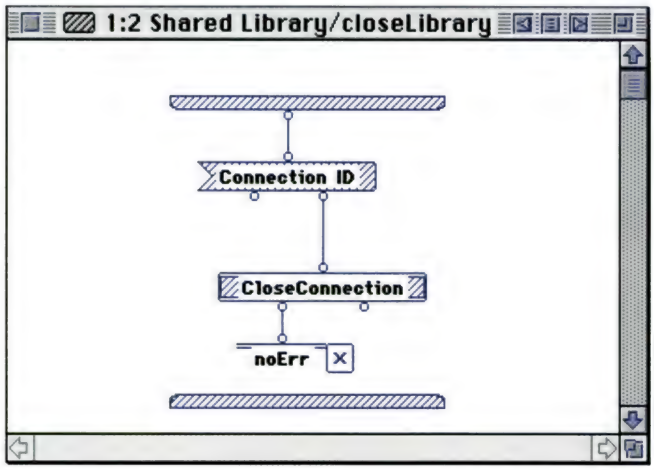



*Figure 6. /closeLibrary closes the connection to the shared library*

### /executeFunction

This method calls the function in the shared library (Figure 7). Two `Shared Library` methods are called prior to the final call to the shared library function via the XPrim `exec-code-frag`. The method `/getGlue` gets a chunk of embedded code (Shackelford [1994]) that `exec-code-frag` needs. The method `/findSymbol` gets a pointer to the function in the shared library. The XPrim `exec-code-frag` makes the call to the selected function.

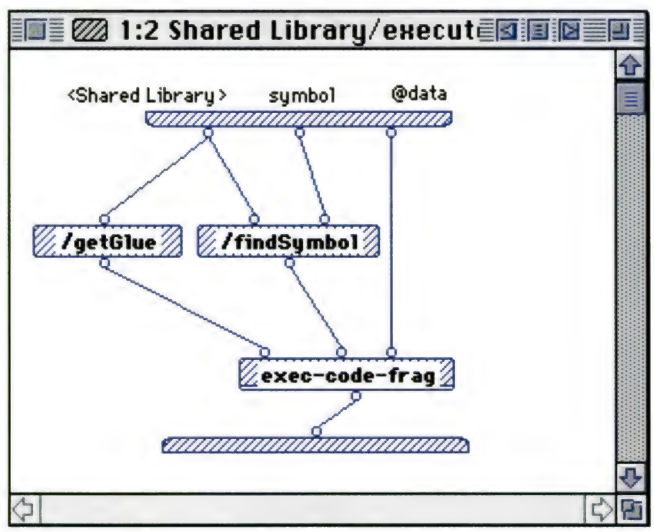
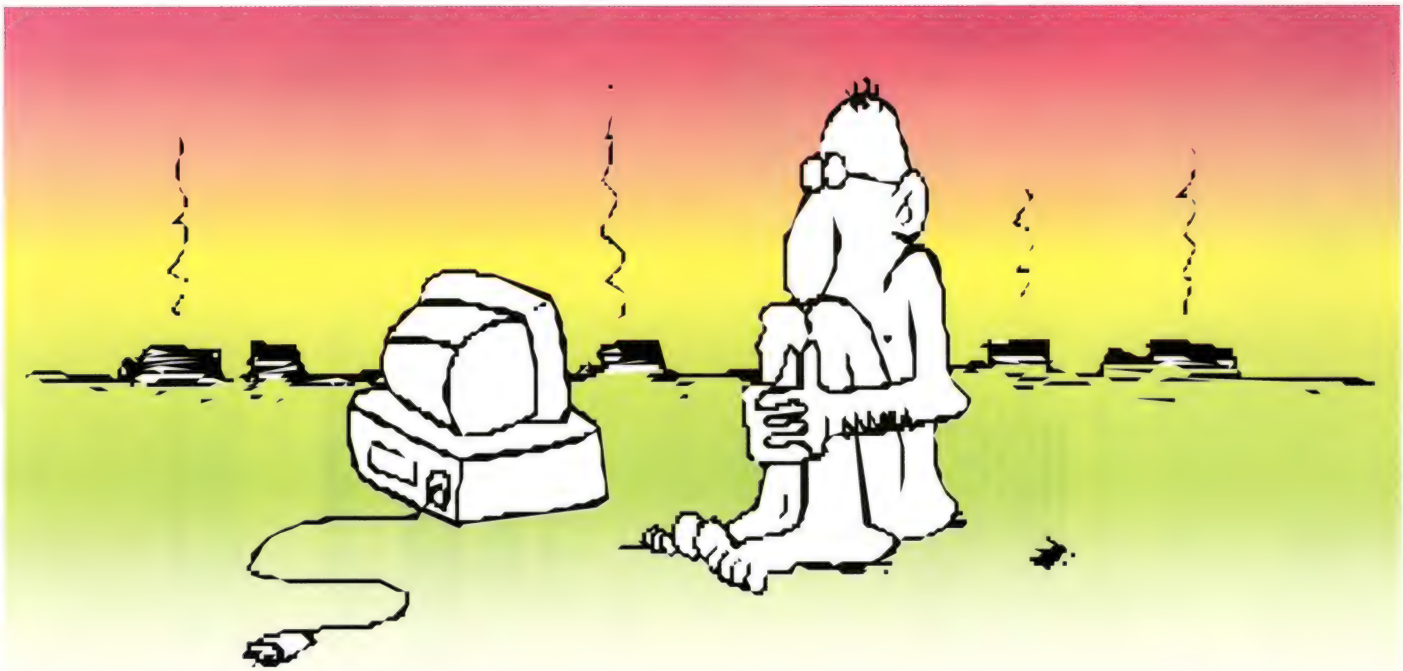



*Figure 7. /executeFunction encapsulates low-level CFM calls*



# Do I have to use a PC to port Windows applications to a Mac?

## WILLOWS TWIN APIW for Macintosh

## Windows API tool box extension for Macintosh developers

- Multi-platform implementation of the Windows API
- Works with Macintosh tools and compilers
- Native library optimized for PowerPC or 68K
- Work within the IDE of your choice

**Call now: (408) 777-1820**

**Visit our web site: http://www.willows.com**

Come and see us at MacWorld!

WILLOWS SOFTWARE

**Willows Software, Inc.** 12950 Saratoga Avenue, Suite A, Saratoga, CA 95070

All trademarks are trademarks of their respective companies.

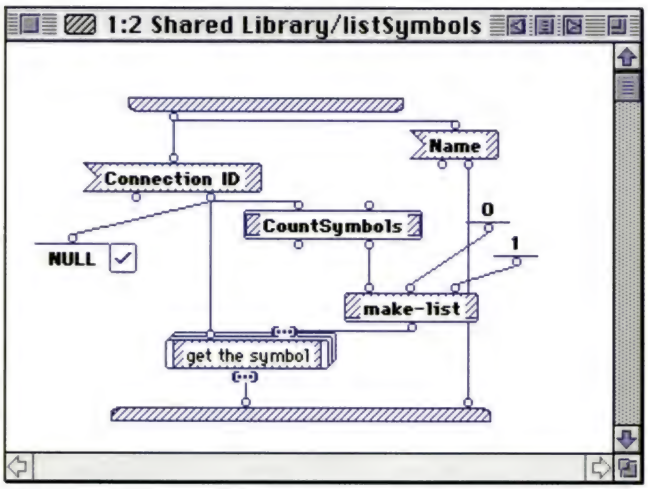



*Figure 8. /listSymbols obtains a list of function pointers*

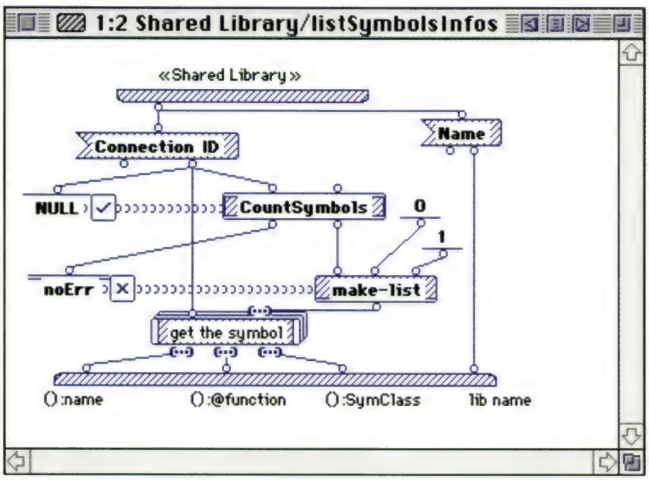



*Figure 9. /listSymbolInfos gets basic information for symbols exported by a shared library*

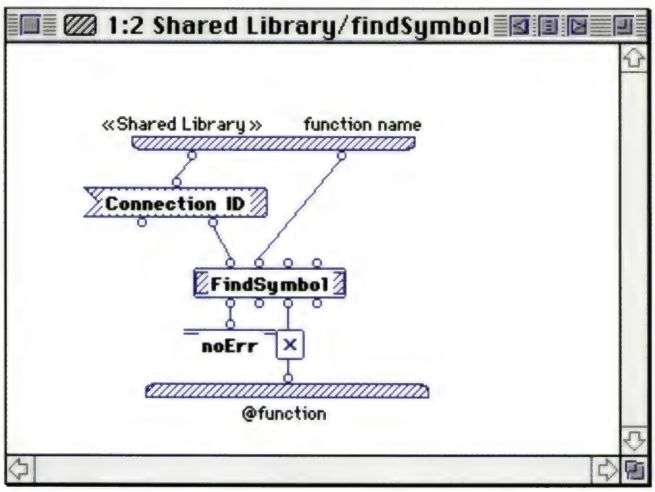



*Figure 10. /findSymbol gets the function pointer for a specific symbol by name*

### /listSymbols

This method can be used to get pointers to the functions exported by the shared library (Figure 8).

### /listSymbolInfos

This method can be used to get pointers to the functions exported by the shared library, names of the functions, and SymClass for each function (Figure 9).

### /findSymbol

This method can be used to get a specific function pointer. You must pass in the name of the function (Figure 10).

## C++ Wrapper

For this article I created a small C++ library. I'll show a function in that library and the wrapper code that was written to access it from CPX. The wrapper technique is basically the same for each function. I won't discuss the pure C library (ExampleCLib), since the C++ version is more general as concerns the techniques required to access shared libraries from CPX. The C++ code is a simple class called `Tvector`. It provides a way to do calculations on a single-dimensional array of data. The next few listings show the declaration file for the shared library (OOSharedLib), the declaration file for the data structure we use to pass data into the shared libraries from CPX, the declaration file for the wrapper functions, the declaration file for the `Tvector` class, and the implementation files for the wrapper functions and the `Tvector` class.

Listing 1: OOSharedLib.h

Declaration file for the OOP shared library for this article.

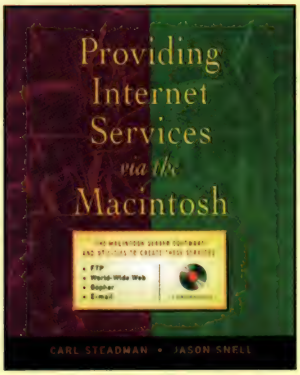
```
#ifndef OOSHAREDLIB_H
#define OOSHAREDLIB_H
/*
   OOSharedLib.h

   Purpose: Header file for OOSharedLib.c

   This is the public function interface.
*/

#include "OOSharedLibStructs.h"

#if defined(__cplusplus)
extern "C" {
#endif


void get_average(data_cache *data);
void get_maximum(data_cache *data);
void get_minimum(data_cache *data);
void get_sum(data_cache *data);
void get_variance(data_cache *data);

#if defined(__cplusplus)
}
#endif
#endif
```

Listing 2: TanLibStructs.h

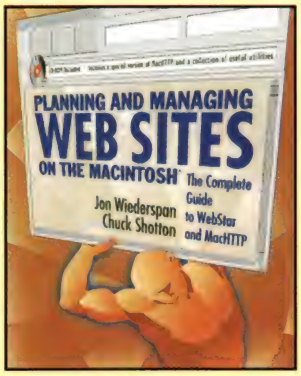
Declaration file for the data_cache. This is in a separate file so that we can use it with Prograph C Tool to "expose" or define the data structures for CPX - i.e. we can allocate the structure and fill the fields.

```
#ifndef TAN_LIB_STRUCTS_H
#define TAN_LIB_STRUCTS_H
```

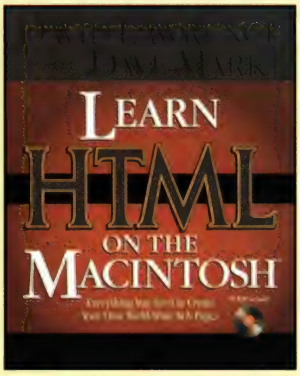
# NEW BOOKS FOR MACINTOSH DEVELOPERS FROM ADDISON-WESLEY BROUGHT TO YOU BY Developer DEPOT




by Carl Steadman
and Jason Snell
Foreword by Peter Lewis
You don't need to be a UNIX guru to offer Internet services. This book shows you how to provide Web pages, FTP, gopher, e-mail, and more with the Mac OS. A companion CD-ROM contains all the Mac server software and utilities you need.
432 pages w/CD-ROM
0-201-48998-8



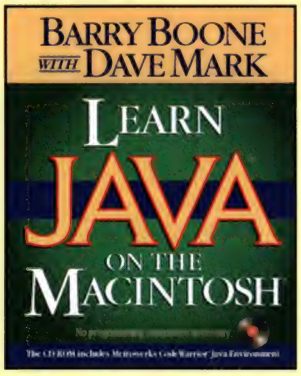

by Jon Wiederspan
and Chuck Shotton
This book is a definitive guide to setting up and running a Web site on the Macintosh. It skillfully teaches you everything you need to know about using WebSTAR, the best known Mac HTTP server software and its shareware predecessor MacHTTP, as well as programming guides for writing CGI applications for your server. A special version of WebSTAR, plus tons of useful software, are on the CD-ROM.
384 pages w/CD-ROM
0-201-47957-5




by David Lawrence
and Dave Mark
HTML is the first thing that anyone who wants to post Web pages needs to know. Bestselling author Dave Mark and Webmaster David Lawrence show you how to use HTML 3.0 with the Macintosh tools on the accompanying CD-ROM to design stunning multimedia Web pages.
304 pages w/CD-ROM
0-201-88793-2

by Tony Francis
*Mac OS 8 Revealed* is the first authoritative look at this exciting new operating system. The companion CD-ROM contains demonstrations of the new features of Mac OS 8. This book is a must for Macintosh developers who want to make their software compatible with Mac OS 8 and for system administrators who need to plan for the upgrade of their systems.
352 pages w/CD-ROM
0-201-47955-9

by Apple Computer, Inc.
Apple's official reference is a must for developers who want to make customizable Internet access available to all Macintosh users. The accompanying CD-ROM contains all the development tools needed to create Cyberdog-aware components.
416 pages w/CD-ROM
0-201-18375-7




by Barry Boone
with Dave Mark
This easy-to-follow introduction takes beginning programmers and Webmasters through the core concepts of this new programming language, including object-oriented techniques, unique Java features such as garbage collection, and basic programming concepts such as working with variables, threads, and classes. The authors also show how to apply this knowledge to write original Java applets for Web pages.
496 pages w/CD-ROM
0-201-19157-1

## ORDER THESE BOOKS FROM DEVELOPER DEPOT NOW!!

| Title | Retail Price | Developer Depot Special Discounted Price |
|---|---|---|
| Providing Internet Services via the Mac OS | $34.95 | $31.46 |
| Plan. and Manag. Web Sites on the Mac | $39.95 | $35.96 |
| Learn HTML on the Macintosh | $29.95 | $26.95 |
| Mac OS 8 Revealed | $34.95 | $31.45 |
| Cyberdog Programmer's Kit | $34.50 | $31.05 |
| Learn Java on the Macintosh | $34.95 | $31.45 |

**All of the books at left and MORE, are listed in the Developer Depot at these Special Discounted Prices.**

Developer DEPOT **Order by Mail, Telephone, Fax, or E-Mail at the following addresses:**

P.O. Box 5200
Westlake Village, CA 91359-5200
Voice Within U.S. (800) MAC DEV1/(800) 622-3381
Outside U.S. (805) 494-9797
Fax (805) 494-9798
email: orders@devdepot.com

You may use your VISA, MasterCard or American Express; or you may send check or money order (in US funds only). Don't forget to include your name, address and phone number with each order. (California residents only, add 7.25% tax to your order.)

```
/*
   © 1995 Tangent Systems
   All Rights Reserved.

   Public declarations for interface structures.
   Putting these declarations in a separate header makes it easier when it
   comes time to use Prograph C Tool to define these strcutures for use in CPX.
*/

#pragma pgtype double Real8
#pragma pgtype int Int4
#pragma pgtype  unsigned Nat4

typedef struct {
           unsigned long length;
           int  success;
           double result;
           double *data;
         } data_cache;

#endif
```

Listing 3: Tvector.h

Declaration file for Tvector class. Defines simple methods for doing calculations on one dimensional arrays of doubles. Each Tvector has an array of doubles.

```
#ifndef TVECTOR_H
#define TVECTOR_H

class Tvector
{
   public:
   // Attributes
   unsigned long length;  // Number of element in this object
   double *theVector;

   // Methods
   Tvector(unsigned long length, double *aVector);
   ~Tvector();

   int calc_average(double *result);
   int calc_maximum(double *result);
   int calc_minimum(double *result);
   int calc_sum(double *result);
   int calc_variance(double *result);

   protected:

   private:
   int ok_vector;

};

#endif
```

As you can see, the class provides a constructor, a destructor, and five "mathematical" methods. Let's look at `get_average`, which wraps the method `calc_average`. One limitation we have is that on the CPX side we will be allowed only one argument when making a call. So each wrapper will be designed to provide one argument – a pointer to a structure. We will handle the allocation and deletion of `Tvector` objects as needed.

Listing 4: get_average

Wraps method "calc_average" in class Tvector. Uses the Tvector class to calculate the average value of an array of doubles buried in data_cache.

```
/*
Pass in a pointer to data_cache, calculates the average value and returns
the answer in data_cache->result.
*/
void get_average(data_cache *theData)
{
   Tvector* aVector;
   double average = 0;
   TestDebug;

   aVector = new Tvector(theData->length, theData->data);
   theData->success = aVector->calc_average(&average);

   theData->result = average;
   delete(aVector);
}
```

The C function `get_average` instantiates a `Tvector` (`aVector`), passing the data to it (a pointer to `data_cache`). It then sends the message `calc_average` to `aVector`. The function captures the result in a local variable called `average`. The value of `average` is then passed back in the `data_cache` structure. All the wrappers are written so that there is only one calling argument and it is a pointer to a data structure. Here are the `Tvector` methods used in the wrapper function `get_average`.

Listing 5: Tvector

Constructor, destructor and method "calc_average" for class Tvector. This is only a partial listing of the the file Tvector.c.

```
Tvector::Tvector(unsigned long theLength, double *aVector)
{
   int i;
   ok_vector = 0;
   if(theLength < 0)
   {
      length = 0;
      return;
   }
   theVector = new double[theLength];

   for(i = 0;i < theLength; i++)
   {
      theVector[i] = aVector[i];
   }

   length = theLength;
   ok_vector = 1;
}

Tvector::~Tvector()
{
   delete(theVector);
   ok_vector = 0;
}

int Tvector::calc_average(double *result)
{
   int i;
   double theResult = 0;
   *result = theResult;
   if(ok_vector == 1)
   {
      for(i=0; i<length; i++)
      {
         theResult += theVector[i];
      }
      theResult = theResult/i;
   }
   *result = theResult;
   return ok_vector;
}
```

### Building Shared Libraries

I used CodeWarrior for this article, and now I'll show the files you'll need to create a shared library. In general there are some special files you'll need to include in the project file, there are some preferences you'll have to set for the project,

Your company has spent millions of dollars developing your software.

# NOW GIVE IT AWAY FOR FREE

**Built-in salesperson**
If software is so smart, why can't it sell itself? With A/Pay, it finally can.

Have we lost our marbles, telling you to give your software away for free? Not really. In today's market, this "absurd" strategy makes sense… *if* you can guarantee that you'll collect money after users have had a chance to try your software.

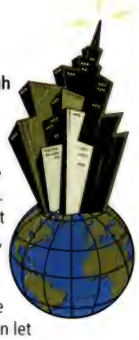
## Put a salesperson inside your software

A/PAY™ is a software module that acts like a built-in salesperson. Windows or Mac developers can easily embed A/PAY in their software. Once you've installed A/PAY, you can distribute your program through the Internet without worrying about receiving payment in advance. Give your software away for free from your Web site, e-mail it to potential users, give it away on free trial disks or CD-ROMs—do whatever it takes to get your program in people's hands.

## Let people try it...and make them buy it

A/PAY lets you control access to your program, so that users can try your software for a limited period of time, or try limited features. At an appropriate time, A/PAY lets the user pay you instantly at the touch of a key. Users can pay securely through the Internet or with a modem. A/PAY also supports 24-hour phone service, fax support, and mail orders. Once the user pays, the software is instantly unlocked and fully usable.

**"Try-before-you-buy" software**
How many potential users are out there, unwilling to risk buying your software? How many of your competitors are adopting a "try-before-you-buy" strategy? Try-before-you-buy is the best way to acquire new users, gain market share, and build a software empire.



**Distribute through the Internet**
Once A/Pay is installed in your program, distribute it however you like. E-mail it, give it out from your Web site, do whatever it takes. Just make sure it gets into the hands of users. Then let your software sell itself.

## What if people copy your software?

Because the A/PAY salesperson is built into your program, it goes wherever your software goes, and is always ready to make a sale. If a user pays for your program, but then gives a copy to his friend, A/PAY can recognize this and can be programmed to collect payment from the new user.

## We take care of everything—so you don't have to

Just plug A/PAY into your program and you're ready to go. Release Software Corporation runs the back-end payment verification network, the 24-hour voice telephone system, merchant account services, and the fax and mail processing center. All you have to do is install the A/PAY salesperson in your code, call us, and release your software. We take care of everything else.

## Want a free demo?

Intrigued? Call us at (415) 833-0200 and request our free A/PAY SDK or visit our web site at www.releasesoft.com and download a free demo.

RELEASE
SOFTWARE CORPORATION

Release Software Corporation 200 Middlefield Road, Suite 220 Menlo Park, CA 94025 USA 415-833-0200 info@releasesoft.com www.releasesoft.com

© 1996 Release Software Corporation. All Rights Reserved. A/PAY and Release Software are trademarks of Release Software Corporation.

"DATA 'FKEY' (6) {
$"514F A924 A873 3F3C"
$"BF86 A9BC 2057 2050"
$"4868 0002 A8F6 4E75"
};"

# 56 million people have developed a passion for Macintosh. You, however, may end up developing something more.

Macintosh users have always been passionate about their computers. In fact, 90% of them buy Mac® computers again when choosing a second system—and 39% actually call themselves "missionaries" for the brand.

Bottom line: When it comes to customer loyalty and satisfaction, no other brand even comes close. People who use Macintosh, quite simply, love it.

Which explains why, in 1995, we sold more Apple® Macintosh® computers to more people than ever before.

## Welcome to Apple Developer Relations

The reason so many people love what they can get out of Macintosh is that developers like you put a lot into it.

And what you can put into it are ideas—big ideas that become distinctly superior customer solutions.

To help you turn your big ideas into reality, Apple provides an extensive library of resources.

Our World Wide Web site includes technical strategies and documentation, software developer kits, on-line training, and marketing information.

For those who want more, we also offer a wide range of development tools, classroom training, and support programs.

And once your product is complete, we can help you reach millions of customers.

Passionate customers.

So whether you're a newcomer to Macintosh or an experienced Mac developer, there's a good chance we'll have exactly what you're looking for.

## Your Success Is Our Commitment

It's important to us that you succeed, because we know your ideas can make a difference.

That's why our commitment to developers is so strong, and why we offer as much support as we do.

So take a few moments to visit our Web site today. Or call for a free copy of the Apple Developer Catalog.

And see for yourself how much more there is to Macintosh.

Contact us at 1-800-282-2732 or, http://dev.info.apple.com.

Apple Developer Relations

Outside the US, call: Australia +61 2 452 8000 Belgium, Luxembourg +32 2 4744290 Brazil 011 5511 886 8000 Asia 852 2506 8888 France 05 90 72 26 Germany, Austria +49 89 99640 534 Italy +39 2 273 26 1 Japan +81 3 5717 7520 Latin America +1 305 569 2759 The Netherlands +31 30 6593890 Nordic Region +46 8 703 30 99 Spain +34 1 663 17 80 Switzerland +41 1 877 9119 United Kingdom, Ireland 0800 50 50 94, +44 131 458 6736 Canada +1 800 637 0029. If you are located in Africa, Central Europe, Eastern Europe, the Mediterranean, or the Middle East, please call your local Apple representative, or look on our web site for further information.

© 1996 Apple Computer, Inc. All rights reserved. Apple, the Apple logo, and Macintosh are trademarks of Apple Computer, Inc. registered in the U.S. and other countries. Mac is a trademark of Apple Computer, Inc.

and then there is an export definition file (the `.exp` file) that you'll need to define. According to the CodeWarrior documentation, you'll need the files listed below under "libraries" and "MW Sources" (Figure 11).

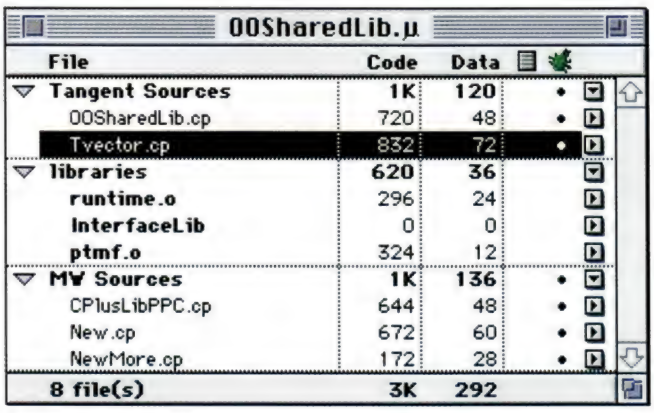



*Figure 11. To build a shared library there are a few required files*

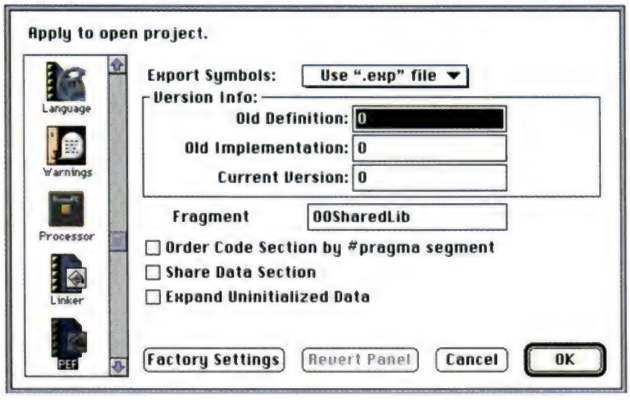



*Figure 12. Define the name of the shared library in the PEF preference panel*

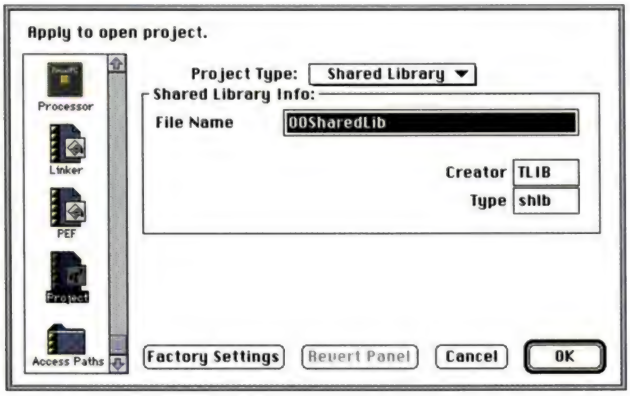



*Figure 13. Define the creator and type attributes of the shared library in the Project preference panel*

Before building the shared library there a few preferences you'll need to set up. See the CodeWarrior documentation for a complete description of the steps. I'll highlight a few items to get you started. You'll have to define the name of the library or fragment. That is defined in the "PEF" portion of the preferences in CodeWarrior (Figure 12).

You'll also define the type and creator attributes, and the fact that the project is going to build a shared library. This is accomplished in the "Project" preferences panel (Figure 13).

One of the final steps before building the shared library is to define the symbols to be exported. If the Project file does not find the appropriate `.exp` file, it will export all functions. You can control what functions are made publicly available by creating a file with the same name as the project file, plus the `.exp` extension. When compilation occurs, the exported symbols will be those functions listed in the `.exp` file. So, for example, the project file for the OOSharedLib is called `OOSharedLib.μ`. The export file for this project would be named `OOSharedLib.μ.exp`. Listing 6 shows the content of the file in order to export our functions.

Listing 6: Export file

The export file defines the symbols that are made available for external use.

```
get_average
get_maximum
get_minimum
get_sum
get_variance
```

## Conclusion and Future Plans

With the information in this article, CPX developers can now start planning, designing, and experimenting with shared libraries on the Power Mac. This powerful new feature of the Mac OS offers exciting possibilities to CPX developers, and the code in this article provides a mechanism to unleash that power.

We view the SLD Kit as another tool for CPX developers to use. We are developing a suite of high-performance technically-oriented shared libraries. The first commercial shared library we are releasing for CPX developers will be a Wavelet packet library. The core of the library is written in C++ and currently runs on DOS, Windows and UNIX. A CPX class will be developed to encapsulate the calls to the shared library. The class will provide basic data compression and Wavelet packet analysis functions. Potential application areas for such a library include voice compression, speech recognition, and digital communications.

## Bibliography and References

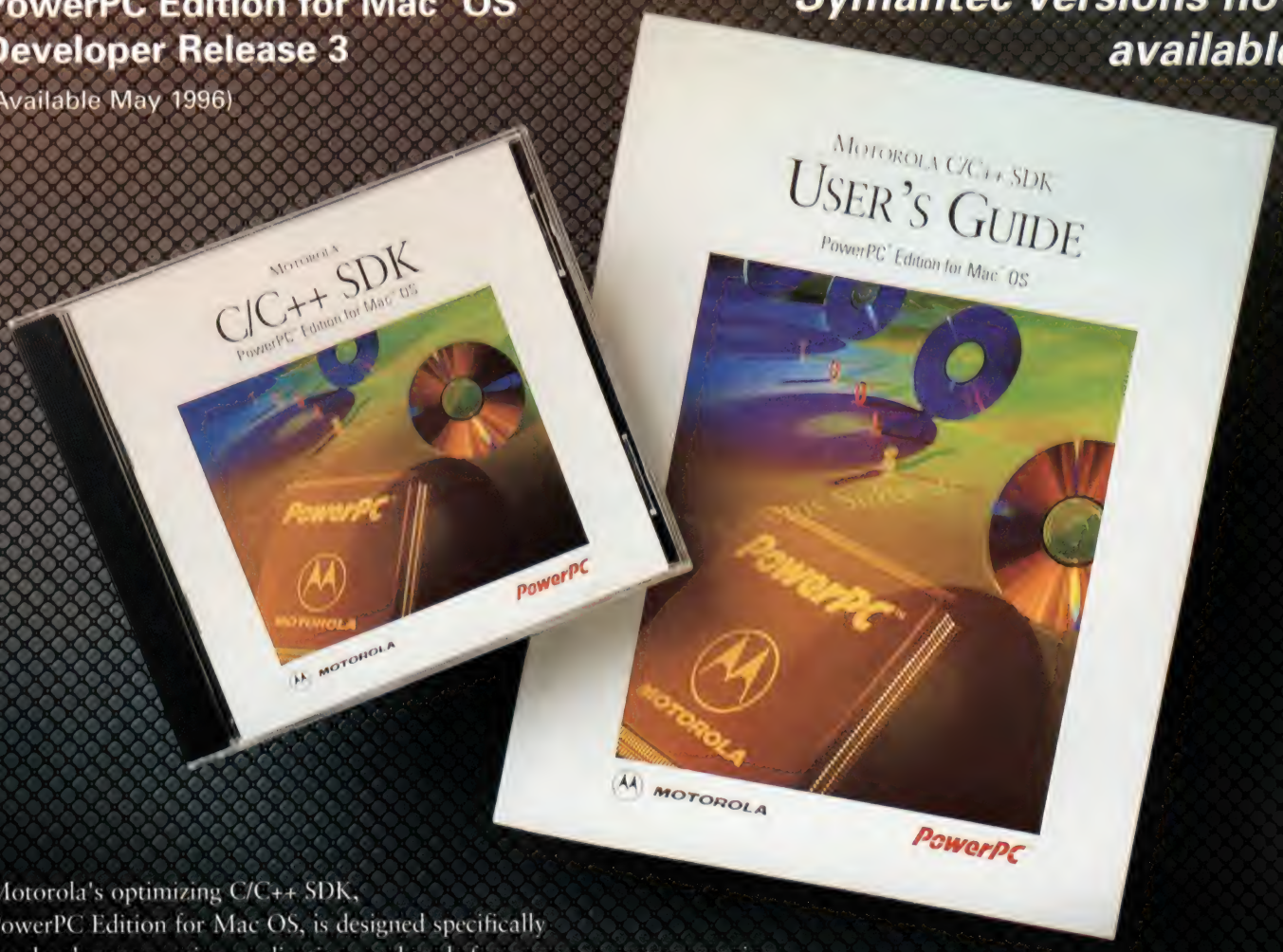
Shackelford, John H. "You Can Have Your 'C' and Call It Too!", *Visual News* 15 (November 1994) 9–12.

# Unlock the power of PowerPC™ microprocessors.

## Motorola C/C++ SDK

**PowerPC Edition for Mac™ OS**
**Developer Release 3**
(Available May 1996)

***Codewarrior and Symantec versions now available!***




Motorola's optimizing C/C++ SDK, PowerPC Edition for Mac OS, is designed specifically for developers creating applications and tools for computer systemsrunning Mac OS. The Motorola compiler works with Metrowerks CodeWarrior, Symantec C/C++ and Apple Macintosh Programmer's Workshop (MPW®) to offer a powerful set of development tools within a complete development environment.

The SDK features the Kuck & Associates Preprocessor (KAP™) — a powerful code restructuring tool designed to improve performance across a broad range of application programs. The Motorola C/C++ SDK delivers world-class code optimization for any specific PowerPC microprocessor, while ensuring compatibility across the entire PowerPC family of microprocessors.

**Call 1-800-347-8384 or 1-512-891-2999.**

**Internet: ppcinfo@risc.sps.mot.com** **http://www.mot.com/PowerPC/**

© 1996 Motorola, Inc. All rights reserved. Motorola and Ⓜ are registered trademarks of Motorola, Inc. PowerPC is a trademark of IBM Corp. & is used under license therefrom. All other trademarks belong to their respective owners.

PowerPC™ MOTOROLA

# Best of Both Worlds

Work on a world-class Macintosh® development team in San Jose, California with the excitement of a start-up, yet the resources and security of the largest software developer in the world.

Who are we? We are a growing team of experienced Macintosh developers,who joined Microsoft eight months ago to build superior internet products.

What are we doing? We're using Metrowerks CodeWarrior and technologies such as QuickTime™, QuickDraw 3D™, AppleScript™, Drag and Drop, OpenTransport, Internet Config, and Frontier™ to build small, fast internet products for Macintosh.

Internet Explorer for Macintosh is just the start. We have other products under development and are always looking for good ideas and great people. If you are a hardcore Macintosh developer driven to build world-class products, this is an opportunity you need to consider. *(whether you code, test, document, evangelize, author, or illustrate).*

If you want to check out our work, go download it for free from
http://www.microsoft.com/ie/iedl.htm

When you're ready for the best of both worlds, send us a resume at:
**Microsoft Employment**
3590 North First Street
Suite 300
San Jose, CA 95134
Fax it to 408 474 3839
email: BayJobs@microsoft.com

**Microsoft®**

An equal opportunity employer. Macintosh, QuickTime, QuickDraw 3D, and AppleScript are registered trademarks of Apple Computer. Frontier is a registered trademark of Userland Software.

*By Marty Wachter*

# Two Java Environments Compared

## *It just started brewing so it might be a little weak*

*Because developments in the Java world refuse to pause for us, with toolmakers rushing to get products into programmers' hands with or without their full complement of functionalities, we are compelled to settle for an occasional description of the situation at some arbitrary moment, knowing full well that we'll be overcome by the rush of events before we go to press. This review was written in mid-May, when, on the Metrowerks side, CodeWarrior 9 (with the corresponding DJ1 update to Discover Programming) had not yet appeared, and, on the Symantec side, neither had the DR2 update to Café – so we couldn't take account of the important changes that those updates would bring. Meanwhile, Natural Intelligence's Roaster DR2 was just on the point of release, and because it involved significant improvements we felt it best to wait for it rather than include Roaster DR1 in the present comparison. We need hardly tell our readers that the only reliable way to stay on top of events is to watch the Web. – man*

http://www.symantec.com
http://www.metrowerks.com
http://www.natural.com

It's about time! Java development tools from the familiar Mac development tool vendors, Metrowerks and Symantec, are finally here. Speculation about their arrival first started showing up in *MacTech Magazine* and the Internet newsgroups almost a year ago. Was it worth the wait? You can decide for yourself by the time you are finished reading this review, which deals with Metrowerks' first release of **Discover Programming with Java** and Symantec's DR1 release of **Café**.

Note that these are Java development tools; this article is not about the Java language. *MacTech Magazine* has been describing the language in detail for some time now; there was, for example, Richard Cardona's "Writing Java 102" in the March 1996 issue ([12.3] 25–38), and, since May, Dave Mark has been devoting some "Getting Started" columns to Java. You will doubtless want to watch Sun's Java Web site at http://java.sun.com, and the many other fine Java sites such as Elliotte Rusty Harold's http://sunsite.unc.edu/javafaq.

Being the first commercial release, both products have numerous limitations that I will outline below; remember that by the time you read this, Metrowerks and Symantec will have updates to their tools available on their Web sites.

The best way for me to approach these products is to describe and compare both tool sets from installation to execution.

### What You Get

Both products include source code editors with syntax coloring, a Sun-validated Java compiler, Java Applet Viewer, some form of class browser, support for code control, all of the Sun Java example projects converted to the appropriate environment, vendor-specific demo projects, Sun's Java documentation in HTML format, and various other forms of documentation and utilities.

The MW product also includes the CodeWarrior Java IDE, which is the full version of their IDE, easily adapted to support

***Marty Wachter***, a freelance Macintosh developer and Internet explorer, is the author of the award-winning MacErrors program. He can't wait for the day when all Java development is done on the Mac. Along with his wife and two cats he enjoys watching movies solely for their surround-sound content. He can be reached at mwachter@ubmail.ubalt.edu.

Java via a plug-in compiler, linker, and preference panel. A big plus for the Metrowerks package is the inclusion of the MW Debugger, which supports the source-level debugging of `.class` files in the same way it supports debugging `.SYM` files for C and C++. Additional documentation is supplied as cool Apple Guide files and interactive help which is also linked to the Java HTML documentation via hypertext links in the Guide files. The entire tool set runs native on 68K and PowerPC platforms.

The Symantec product includes the Symantec Project Manager (which appears also to be the full version) for source editing and project management, and a very preliminary version of a visual GUI development tool called Café Studio .which is similar in concept to their Visual Architect tool, for creating `java.awt` GUI elements and generating the source code for the interface.

In addition to the documentation on the Metrowerks disk, they include an electronic copy of the book *Learn Java on the Macintosh*, by Barry Boone. Symantec strikes back with an electronic copy of their own "Intro To Java Programming". Although both packages have ample documentation, I really would have preferred the Java API docs in either THINK Reference format and/or QuickView format.

### What You Don't Get

In their current releases, both products are missing the promised JIT (just-in-time) compilers that their respective documentation alludes to. I suspect that these compilers will greatly enhance the usability of both tool sets.

Other than the absence of a GUI tool for creating Java AWT interfaces, the Metrowerks product contains everything you need to start developing your own Java applets and applications. I don't know if such a tool is planned, but extending their PowerPlant interface builder would be a welcome addition.

The Symantec tool, on the other hand, is missing two huge features in its DR1 release. If you don't have a Power Mac, you're out of luck until the 68K tools are available. The lack of any kind of source-level debugger will make this release very difficult to use for learning the language and for commercial Java development. JPEG support as well as sound support are also absent from this release.

### Requirements

According to their documentation, the Metrowerks tools require the following minimum configuration:

- A Mac OS computer with an MC68020, MC68030, MC68040, or PowerPC processor
- 8 megabytes of RAM
- Mac OS System Software 7.1 or later (you'll need 7.5 or the Apple Guide system if you want to use the Apple Guide documentation)
- A CD-ROM reader for installing the software

For optimum performance, Metrowerks suggests an MC68040 or PowerPC machine with 16 megabytes of RAM.

Symantec minimum requirements are as follows:

- Macintosh 68030 processor (or greater) or Power Macintosh (DR1 supports only PowerPC development)
- 16 megabytes of RAM
- CD-ROM reader for installation

The Symantec documentation did not offer any optimum configuration, but ultimately the only way I could get an applet up in the SPM and run it on my Power Mac 6100 (System 7.1.2) with 24 megs of RAM, was to turn off all extensions and turn on Ram Doubler to get it up to 48 megs.

### Installation

So, here I am, about to install both products. They each ship on a CD-ROM and contain installers built by MindVision's Installer Vise system, with options for Easy and Custom installations. Metrowerks' recommended Easy Install option will need almost 51 megabytes of your disk. Thankfully, Metrowerks gives you the option of installing the tools, documentation, online books, and sample code separately. There is also the comforting Minimal Install option which will cost you under 11 megabytes of space.

The Symantec installer provides an Easy Install option, but it doesn't tell me how much space it will cost me. Only after performing the installation will I learn that Easy Install requires about 35 megabytes of space (5 megs more than the required space according to the documentation). The Easy Install option does not install the full documentation, the extra utilities, or the Café Studio tool. To get around this, you are offered the option of installing the Café Tools, Café Studio, Java Demos, Goodies, and full documentation under the Custom Install choice. Selecting all of the Installation options under Custom Install reports that the full installation will require just under 28 megabytes of space – I'm confused!

I proceed with the Easy Installation options for both packages. The Metrowerks system installs everything into a "Discover Programming with Java" folder on my drive. The Symantec installer is a bit troublesome in that the default installation places the Café tools folders at the root level of my drive, so I have to do some cleaning up after the install. Sure, I forgot to select a directory in the "Select a Folder" menu under the Install Location option, but I was excited to get started. The Easy Install should be just that, easy. I then have to go back and do a Custom Install to get the Café Studio tool.

### The Tools

OK, now that I finally have these tools on my Mac, it's time to see what it's all about. Generally speaking, both environments are extensions of their respective C/C++ products. This is fortunate because you don't have to learn new tools to get started. To keep things even, I chose the Sun Animator demo applet, included with both packages, for my tests.

### IDE Project Managers

As you would expect from any successful Macintosh tools vendor, both products contain a familiar project model for building Java applets and Java applications (applets are executed via the `<applet>` tag in HTML; applications are real Macintosh programs that execute when you double-click them). Symantec includes its standard Symantec Project Manager for managing your `.java` source files and your `.html` files; Metrowerks uses their CodeWarrior Integrated Development Environment for the same purpose.

### Editors

Both products include editors built into their IDEs, which support function markers, syntax coloring, drag-and-drop editing, and split panes.

### Class Browsers

Both products also contain class browsers which are extremely helpful for viewing and editing any object-oriented source code. The Symantec product includes a graphical mode which shows you a left-to-right graphical tree of the project's files. This is a cool tool, but I was easily able to cause a few problems while testing it out. Aside from a few cosmetic bugs like list selections extending beyond the bounding panel, I found that if I double-clicked on a Java library class function the SPM reports that the file can't be found. I tried removing the project's object code, and reset the project paths, but this caused a total system crash. After rebooting, I tried again and this time crashed the SPM by holding the mouse down over one of the graphical leaves in the class browser. Oh, well; it looks like a cool feature. On to the rest of the tools.

### Compilers

As I mentioned before, the JIT compilers are not available in either release. Metrowerks currently offers only one compiler, the Sun-validated "MW Compiler" which is essentially the same compiler code that the Sun Mac JDK (beta 1) uses. Symantec offers the standard Sun compiler and a faster Symantec Java Compiler that seems to outperform the Sun compilers as far as time to compile is concerned.

### Interface Builders

Metrowerks doesn't include any sort of interface builder with this release; Symantec, however, has included a preview release of their Café Studio application (which is also PowerPC-only code). This tool reportedly allows you to draw the `java.awt` interface components with a MacDraw-like tool palette. You can the generate the Java code to create the actual interface in your project. You still have to do some manual work on the resulting code, and several of the `java.awt` components are not supported. There is also no way of subclassing any of the `java.awt` classes using this tool. I also found out that there is no documentation for Café Studio on the DR1 disk. You had to visit Symantec's password-protected Café Web site, armed with your user ID and password (which comes on a paper insert to the CD's jewel box) just to get to the Café Studio update, which I found to contain the missing documentation to Café Studio, in Adobe PDF format.

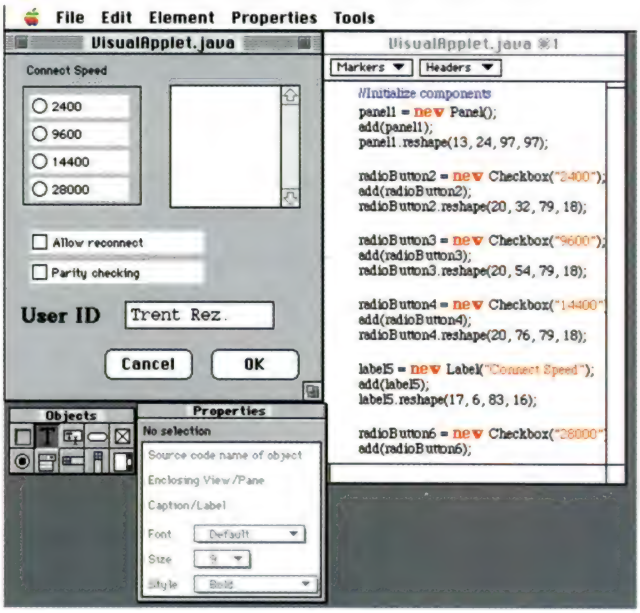



*Figure 1. The Café Studio environment*

### Applet Viewers (The Java Virtual Machine)

Both Metrowerks and Symantec include a modified version of the Sun Applet Viewer. Once your Java source code is compiled into bytecodes, you run your code with the Applet Viewer. The Metrowerks version, called "Metrowerks Java", is a bit farther along in that it supports sound. The Symantec version doesn't support sound yet.

I have found that the overall performance of both Applet Viewers is lacking. I sincerely hope that the next release of both products will pay more attention to the performance of Java on the Mac platform. In particular, the graphics drawing implementations are very slow. I have some custom graph-paper code (similar to the Graphing Calculator that came with the Power Macs) which I wrote on a Sparc 4 under Sun's Solaris JDK, that flies when run with the Sun JDK, but is painfully slow even on a Power Mac 9500/132!

### Debuggers

This part of the review was the easiest. The current release of Café doesn't have one and the Metrowerks debugger is the exact same "MW Debug" that you use with any other language. You just treat your `.class` files that the MW compiler spits out as if they were `.SYM` files. Having a real Mac-based source-level debugger for Java code is the one thing that I have really wanted since I started doing Java development last year with the Sun JDK alpha 3 tools. This tool alone is worth the entire cost of the Metrowerks tool set!

Movie Cleaner Pro
The Right Tool for Video Compression
If your project includes QuickTime movies, you need Movie Cleaner Pro, the most powerful and easy-to-use video compression tool available. Now you can quickly compress movies for CD-ROM or the Web — with unparalleled quality and speed!
Features include:
Drag & Drop Batch Processing - easily create a batch of up to 2,000 movies in minutes.
Automated IMA audio compression and high quality audio resampling for great sound.
Suspend and Resume - stop compressing and then flawlessly resume - even for batches.
Advanced Settings - easily adjust data rates, A/V fades, high-quality frames, and more.
Adaptive Noise Reduction filter, high quality sine image scaling, and much more!
Terran Interactive
Movie Cleaner Pro 1.2
$189.95*
More info and Orders:
www.Terran-Int.com
1-800-577-3443
International (408) 278-9065
* US Dollars + S&H Source code - MTK 0-496

[*Incoming Editor-in-Chief Will Iverson adds: "As of this writing, all three vendors have shipped debuggers. However, all of these debuggers have severe limitations with regard to multi-threaded debugging, variable display, stepping in and out of code, and multi-language display (e.g. debugging C/C++ and Java code). Consider yourself warned..."*]

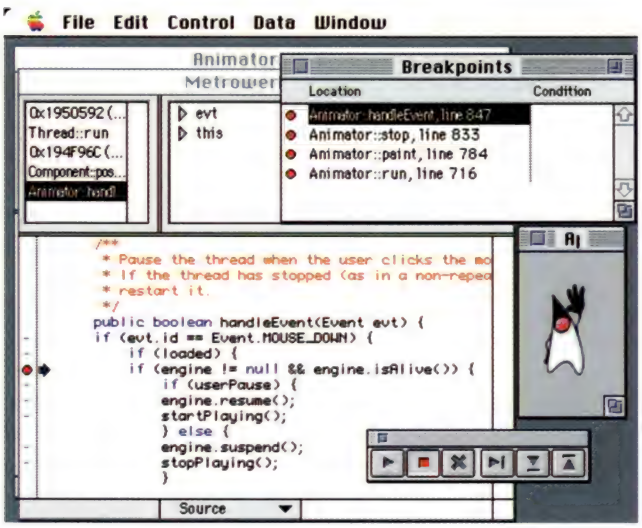



*Figure 2. The Metrowerks debugger*

### Support

The support you get with both products differs a bit. Metrowerks and Symantec both offer some type of phone support, fax support, and internet support. The big differences are that Symantec offers a 60-day money-back guarantee and 90 days (from the date of your first call) of unlimited phone support for installation, configuration, and general use (which in my opinion you'll definitely need), after which you've got to pay, while Metrowerks offers their standard unlimited phone support. The other thing, which I mentioned earlier, is that you'll need your password and user ID to access Symantec's Café Web site for updates, while Metrowerks' Web site is accessible without a password or ID.

### Conclusion

Table 1 shows a summary of the features sets of the two environments.

| | Symantec Café | Discover Java |
|---|---|---|
| Price | $299.95 ($99 intro) | $99 |
| Class Browser | Yes (graphical) | Yes |
| 68K Support | Not Yet | Yes |
| PPC Support | Yes | Yes |
| Visual Tools | Yes (limited) | No |
| Sample Code | Yes | Yes |
| Online Documentation | Yes | Yes |
| Source Control | Yes | Optional |
| Integrated Project Manager | Yes | Yes |
| Colored Syntax | Yes | Yes |
| Source Level Debugging | Not Yet | Yes |
| Support | 90 days | Unlimited |

*Table 1. Comparative feature sets*

It's nice to finally have some options for Java development on the Mac platform. Finally the Internet world will get to see what interesting and useful Java software will look like from the Mac developer's perspective. Your choice of Java tools might come down to going with whatever environment you are more comfortable with, just to ease the learning curve, but in the early innings of this game the smaller hardware requirements, 68K support, and the excellent source-level debugger make Metrowerks' "Discover Programming with Java" the clear choice for getting started with Java development on the Mac. However, if you are planning to do any serious Java development, and especially if you require use of Java's `awt.graphics` system, you'll want to get yourself a Sparc station.

**Visit MacTech Magazine's Web site!**
http://www.mactech.com

*By Rick Hartmann*

SYMANTEC.™

***This monthly column, written by Symantec's Technical Support Engineers, is intended to give our readers technical information on using Symantec products.***

With all the excitement about Java, and the recent release of Symantec Café for Macintosh, we thought we would devote this issue of the Symantec Top Ten to Java.

(As this article goes to press there are updates to Café and Sun's Java Development Kit for Macintosh soon to be released. Due to the rapid development cycle of these Java products, there will likely be numerous changes and bug fixes during this phase. Please read the release notes for the latest version of Café to determine if a specific problem has been resolved, and check the Café Web site, http://cafe.symantec.com, for the latest news regarding Java and Café.)

**Q:** Sometimes when I double-click on a C++ project, I get the Café Version of the Project Manager. Why?

**A:** The standalone Café Project Manager will soon be integrated into the Symantec Project Manager along with C++, C, and Pascal. Therefore the creator type was left the same as the SPM. This may cause some confusion until then, so if you have both Symantec Project Manager for C++ and Café, you should get in the habit of dragging project files directly to the Café icon or an alias on the desktop with a name that identifies it as the Café Project Manager.

**Q:** When I try to use `java.net.ServerSocket` or `java.net.Socket`, etc., I am getting a Java runtime error:

```
Can't get watchdog proc
```

What is wrong?

**A:** The Macintosh Java virtual machine does not correctly handle the `java.net` package as of this writing. We expect an update from Sun that will correct this well before you read this. There is currently no workaround.

**Q:** My applet's `mouseMove()` function is getting called continuously, even when the mouse does not move. Why?

**A:** This is an AppletViewer problem. Try dragging the HTML file from your applet project onto a Java-enabled browser instead.

**Q:** My applet behaves differently depending on whether I run it in the AppletViewer or in Netscape. Why?

**A:** There are a good number of quirks with the AppletViewer right now. Both products are beta. As these products mature and reach their final versions, we expect their behavior to be more consistent.

**Q:** Choosing the default font in Café Studio seems to cause a runtime error:

```
java.lang.NullPointerException
    at java.awt.Font.<init>(Font.java)
    at studio.init(studio.java:101)
    at sun.applet.AppletPanel.run(AppletPanel.java)
    at java.lang.Thread.run(Thread.java)
```

How do I fix this?

**A:** Studio generates the following:

```
myItemfont = new Font(null, Font.PLAIN, 9);
```

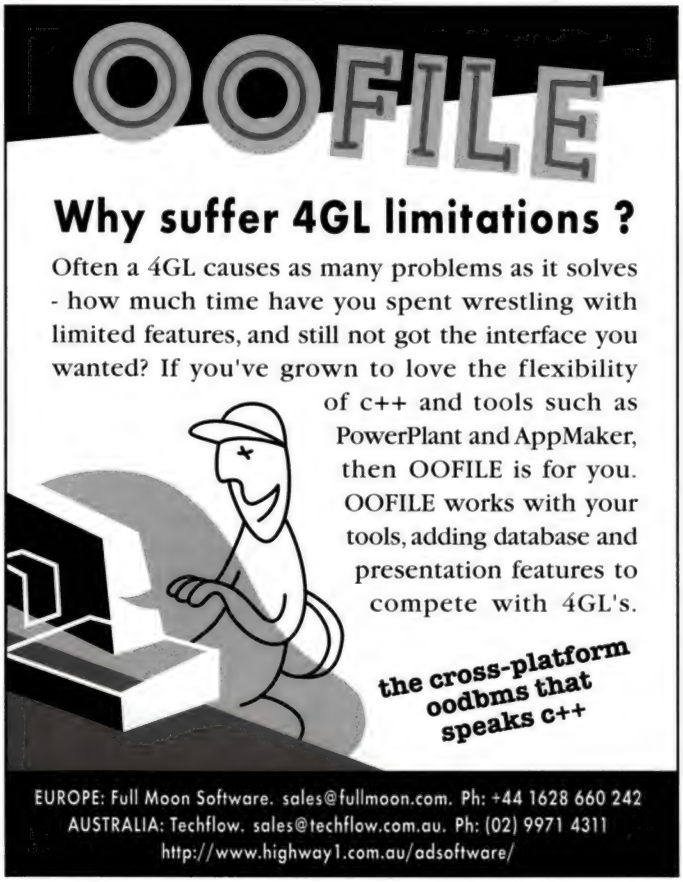
The first parameter must be a `String`; hence the exception. In the next release this will be changed to:

```
myItemfont = new Font("", Font.PLAIN, 9);
```

**Q:** Why am I getting this compile error?

```
Error:  File Foo.class does not contain Foo as expected,
but java.Apackage.Foo.  Please remove the file.
```

**A:** You forgot to import `java.apackage.foo`.


OOFILE
Why suffer 4GL limitations ?
Often a 4GL causes as many problems as it solves - how much time have you spent wrestling with limited features, and still not got the interface you wanted? If you've grown to love the flexibility of c++ and tools such as PowerPlant and AppMaker, then OOFILE is for you. OOFILE works with your tools, adding database and presentation features to compete with 4GL's.
the cross-platform oodbms that speaks c++
EUROPE: Full Moon Software. sales@fullmoon.com. Ph: +44 1628 660 242
AUSTRALIA: Techflow. sales@techflow.com.au. Ph: (02) 9971 4311
http://www.highway1.com.au/adsoftware/

**Q:** I renamed my Java application's `HelloWorld.java` source file to `foo.java`, and also changed `public class HelloWorld` to `public class foo`; but now when I choose **Run** from the Project Manager, I get:

```
Could not complete the last command because the file was
not found... -43
```

or:

```
Could not complete the last command because the AppleEvent
was not handled... -1708
```

What did I forget?

**A:** Under the **Project** menu, in **Options: Project Type**, you must set the "main() class" popup to the name of your class that contains the `main()` method.

**Q:** When I run my Java application, I get this error in the stderr window:

```
java.lang.IllegalAccessException: ClassName
    at sun.applet.AppletPanel.run(AppletPanel.java)
    at java.lang.Thread.run(Thread.java)
java.lang.NullPointerException
    at sun.applet.AppletPanel.run(AppletPanel.java)
    at java.lang.Thread.run(Thread.java)
```

The stdout window contains this message:

```
load: ClassName is not public or has no public
constructor.
```

What is wrong?

**A:** Your project type was set to be an applet instead of an application. Go to the **Project** menu and choose **Options**. Click on the **Project Type**. Set the **Program Type** radio button to **Application** and make sure the "main() class" popup menu is set to your class name that contains your `main()` method.

**Q:** When I try to run the StdinTest program from the Introduction to Java demos, it hangs my system. Why?

**A:** The current version of the Macintosh Java virtual machine is not doing standard (keyboard) input correctly. We expect Sun to correct this soon. It will probably be functional by the time you read this. Check our Web page for the latest information on these issues.

**Q:** How do I save the stdout and stderr windows to a file? Can I redirect stdout and stderr to files, like I can with C and C++?

**A:** Yes to both questions. Launch the AppletViewer. From the Apple menu select the **Java Runtime** menu. It contains menu items for both redirection of stdout and stderr as well as a **Save Text Window** item for the frontmost text window.

**Q:** I drag the applet class file on the AppletViewer and I get an error. How do I run applets?

**A:** Applets are designed to be run within the context of a Web browser that supports the Java virtual machine. You must first create a Web page with the proper HTML tags for your applet. At a minimum, for an applet to run, you'll need:

```
<applet code = myAppletName width=300 height=400>
</applet>
```

You can then drag the HTML file on the AppletViewer or a Java-enabled browser.

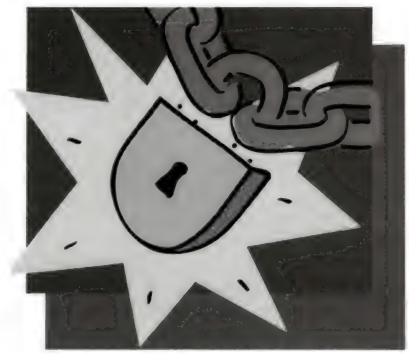
Thanks to Andy McFarland, Kevin Quah, Scott Morison, Noah Lieberman, Michael Hopkins, Levi Brown, Glenn Austin, Steve Howard, Craig Conner, Chris Prinos, Peter Fry, and Mark Baldwin for their contributions to this article.

# dtF The Relational Database System

**dtF v1.6**
including numerous SQL enhancements and ODBC driver available!

When it comes to performance **dtF** is in a class all its own. **dtF** utilizes a proprietary query optimization and caching scheme to obtain unparalleled performance. If you are in need of a quantum leap in performance, use the **dtF** native PowerPC stand-alone or client/server version.

## ...performance



Full transaction control and automatic level-3 error recovery guarantee maximum data protection even after sudden system crashes. **dtF** databases are compressed and encrypted to protect against all unauthorized access, even disk editors. Implement sophisticated table and operation level security with **dtF**'s password features.

## ...data security

## ...ease of use

With **dtF**'s unique direct integration technology, the **dtF** database system is fully contained in your application, enabling you to build double-clickable database applications. Never worry about missing or conflicting INITs or drivers. **dtF**'s native API is compact, easy-to-use and integrates seamlessly with all major development environments on the Macintosh. **dtF**'s high performance SQL, integrated data dictionary, security features, automatic index selection, query optimization and error recovery allow you to concentrate on creating great database applications instead of messing around with the internals of the database system.

**Stand-alone applications built with dtF are royalty-free!**

**dtF** is available for Macintosh System 7.x (68K and native PowerPC). **dtF** supports MPW C/C++, Symantec C/C++, Metrowerks CodeWarrior (all compilers 68K and native PPC). **dtF** is fully OpenDoc™ compatible. Separate versions for use with HyperCard 2.x, SuperCard 2.x, Smalltalk-Agents and Pictorius Peregrine are provided. AppleScript interface via DataScript™ for **dtF** from General Knowledge. **dtF** supports cross-platform development on Windows 3.11, Windows 95 and OS/2.

**dtF Americas, Inc.**
19672 Stevens Creek Blvd.,
Suite 128
Cupertino, CA 95014
USA

Phone: (800) DTF-1790
Fax: (510) 828-8755
AppleLink: DTF.AMERICA
Internet: dtF.america@applelink.apple.com

dtF The Relational Database System

*By Steve Kiene*

# Mac OS 8 Checklist

## *Ensuring your software will work properly*

Copland (now officially restyled "Mac OS 8") represents the greatest evolutionary step in the operating system that the Macintosh has seen in its twelve-year history. It is unrealistic to expect that all software will work with it unmodified. Although Apple's engineers have striven to provide as much backward compatibility as possible, there are several areas which can cause problems for your software. This article will outline what you can do to your software now to ensure that it will run unmodified under Mac OS 8. First, we'll discuss the common types of products and how they will transition to Mac OS 8. Next, we'll talk about the changes you'll need to make to your products to make sure they are compatible with Mac OS 8. Third, we'll list some things you can do to make your product run better under Mac OS 8.

### Common Product Types

The most common types of products fall under the following categories:

**Applications with a user interface**

This accounts for most Macintosh software. Most applications will run unmodified in Mac OS 8. There are, of course, many changes you can make to your application to get better performance in Mac OS 8 (this will probably furnish material for a future article!).

**Faceless background applications**

These are many types of application which lack a user interface, including applications which provide services but do not directly interact with the user. These can be migrated to Mac OS 8 either as a server program or as a Mac OS 8 application. (A server program is a new application type in Mac OS 8; still more material for a future article?)

**Control panels and desk accessories**

These are typically small pieces of code with a single window and a simple user interface. These are *not* supported under Mac OS 8. You should rewrite them to be applications or, if applicable, perhaps as OpenDoc Part Editors.

**Extensions (INITs)**

Extensions are typically used to modify or extend the functionality of the operating system. This type of code is *ignored* in Mac OS 8. Your transition options are to accomplish the same functionality either as a server program or as a regular application. There is a different way to patch Mac OS 8 to extend its functionality – by using the Patch Manager (wait, don't tell me; a possible future article!).

**Drivers (DRVRs)**

Drivers are typically used to communicate with peripheral devices. They fall into two categories: code that touches hardware, and code that doesn't. If your driver touches hardware, you will need to rewrite it as one of the following:

1. A specific Mac OS 8 family plug-in. For example, an ethernet driver such as `.ENET` must be rewritten as an Open Transport networking plug-in.
2. A Device Manager plug-in. This type of driver is known as an `NDRV`, or Native Driver. You can find more information about writing Native Drivers in the document entitled "Designing PCI Cards and Drivers for the Power Macintosh Computer", available from Apple.

***Steve Kiene*** is Copland / Mac OS 8 Contributing Editor to *MacTech™ Magazine*.

# BBEdit 4.0

## now in living color.

**The Tool of Choice for Web Page Designers**

- HTML key-word coloring.
- Full suite of Drag and Drop HTML Tools.
- HTML-aware spelling checker.
- Directly "Open From..." and "Save To..." FTP servers.
- Preview with your favorite web browser.
- Powerful interactive folder/project comparison.
- Frontier 4.0 integration, including valuable scripts to automate common web-authoring tasks.

**The Tool of Choice for Software Developers**

- Source code coloring.
- Integrated support for Symantec C++.
- Integrated support for Metrowerks CodeWarrior.
- Unrivaled search and replace.
- PopupFuncs™ for easy source code navigation. (now supports Java and TeX, as well as C, C++, Pascal, Object Pascal, Rez, FORTRAN, assembly language, tcl, Perl, ScriptX, and AppleScript)

http://www.barebones.com
or
617•676•0650

BBEdit and PopupFuncs are trademarks of Bare Bones Software, Inc. All other trademarks and registered trademarks are properties of their respective holders.
© 1996 Bare Bones Software, Inc.

For drivers which do *not* touch hardware, the requirements are less stringent. Your driver can be modified to be one of the following:

1. A specific Mac OS 8 family plug-in. For example, an AppleTalk protocol module such as `.ADSP` can be rewritten as an Open Transport network plug-in. Drivers written in this way are available for use by both cooperative and re-entrant clients.

2. A Device Manager family plug-in. This option applies to drivers that do not become Mac OS 8 family plug-ins.

3. A `DRVR` using only backward-compatible APIs. This option is not available to any drivers which are part of the page fault path. Any driver which may be called in the path of a page fault must be rewritten to be a Mac OS 8 I/O plug-in.

### Steps To Mac OS 8 Compatibility

To make sure your application at least works under Mac OS 8, you may need to make some changes to your source code. Most of these changes are minor, and Apple has warned about their use before; some of them are a surprise, but at least we're getting some advance warning so we can change our applications now.

1. **Don't use ASLM.** ASLM is not supported under Mac OS 8. You will need to use the Code Fragment Manager or SOM to handle shared library functionality.

2. **Don't access the trap table directly.** Don't make assumptions about the structure of the trap table. Its location, size, and structure are all private. Execution of entries is supported only through A-Traps or through `NGetTrapAddress`. Modification of entries is supported only via `NSetTrapAddress`.

3. **Don't use jGNEFilter to intercepts events globally.** Mac OS 8 has limited support for `jGNEFilter`. The filter will be called only for events intended for the application which installed the filter.

4. **Don't call PPostEvent.** Events put into the event queue with `PPostEvent` will be available only to the application which posted the events.

5. **Don't call GetEvQHdr.** It points to an empty list. Events are maintained on a per-application basis.

6. **Don't access private traps or private LowMem.** All private traps have been removed from the Mac OS 8 trap table.

7. **Don't use PrivateInterfaceLib.** On all current PowerMac machines there exists a code fragment called PrivateInterfaceLib. This contains private routines used by the Operating System. This library does not exist in Mac OS 8.

8. **Don't rely on file system hooks or patches.** `ExtFSHook` and `FSQueueHook` are not supported in Mac OS 8. In addition, patching any of the file system will affect only the application which installs the patches.

9. **Don't use compressed resources.** Resources which are decompressed with the undocumented "dcmp" mechanism in System 7 will not be compatible with Mac OS 8. Resources which are compressed with other third-party mechanisms such as Application VISE will continue to work.

10. **Don't call InitResources, RsrcZoneInit, or RsrcMapEntry.** Both `InitResources` and `RsrcZoneInit` are obsolete and not implemented in Mac OS 8. In System 7, `RsrcMapEntry` was used to read Resource Manager data directly from the resource map. In Mac OS 8, the in-memory resource map is private, and this call is not supported.

11. **Don't use resource file refNums with File Manager calls.** In the past, the Resource Manager used the File Manager to open, close, read, and write the resource fork. Because of this, the refNum returned from `OpenResFile` could be used by an application and passed to File Manager calls to read and write data. In Mac OS 8, the Resource Manager no longer uses the File Manager to read and write the resource fork, so a refNum returned from `OpenResFile` has no relation to a File Manager refNum. You cannot use a File Manager refNum with the Resource Manager, and vice versa.

12. **Don't call SetApplLimit.** In the past you could resize your application stack by calling `SetApplLimit`. Because the stack was directly above the heap, this would shrink the application heap so that the stack could be enlarged. In Mac OS 8, the stack is no longer directly above the heap, so this call is not supported. Use the code fragment resource instead.

13. **Don't use virtual memory calls.** With the microkernel architecture introduced in Mac OS 8, the System 7 virtual memory calls are no longer supported.

14. **Don't assume relationships between the application stack and heap.** As mentioned above, the application stack is no longer located directly above the application heap.

15. **Don't allocate all of temporary memory.** Mac OS 8 implements a demand-paging virtual memory scheme. Because of this, temporary memory is limited only by disk space. Allocate only the amount of memory you need, not all that is available.

16. **Don't allocate all hard drive space.** Because of the demand-paging virtual memory system, disk space is used dynamically as temporary memory. If the disk is full, the system will run out of memory.

Main Event

**The lead widens**

**For professionals,
for novices,
for webmasters,
for solutions providers:**

There's only one serious choice.

**Always the most powerful,
the most feature-packed,
and at the same time,
the easiest to use.**

And it just got better.

**It's already the only tool that lets you watch local variables, trace variables, or change values and fix problems while stepping.**

Now, Scripter hits version 2.0, with still more unique features:

**Handler debugging:** Debug handlers without modifying your scripts

**Applet and CGI simulation:** Interactively debug live interapplication messages sent to a script application or AppleScript CGI from an applet, your Web server or FaceSpan 2.x. You won't have to modify your Web pages or your CGIs.

**Live editing:** Debug script applications and CGIs in their normal environment.

**Object Map:** Explore a graphical view of the class hierarchy of scriptable applications.

**Integrated object database:** ScriptBase, separately $59, is now included with Scripter. Use it to store your data and media elements (frequently used values, scripts, text, pictures, HTML, headers, file references) and share them between scripts. Manage ScriptBase directly from Scripter.

# Scripter® version 2.0

**The authoring, debugging and development environment that's worthy of the power of AppleScript.**

If you're an AppleScript novice, Scripter will take you by the hand and show you the correct syntax for your statements. If you know what you're doing, Scripter will help you do it faster than any other editor. And Scripter is the only tool on the market with *interactive debugging*: You catch the offending code in the act. Only Scripter lets you fix the problem and continue debugging!

We designed Scripter for productivity and user-friendliness. People come to us for Scripter's industrial-strength debugger, but what really impresses them is the speed and ease at which Scripter allows them to work. It's been used to build major corporate process automation systems.

Scripter offers power-assisted statement construction. Vocabulary access in a single mouseclick. Multifunction find and replace. A variable watcher and expression evaluator. Tools to change variable values or try out commands, in context, in the middle of debugging. Automatic navigation to subroutines. Background processing. An enhanced trace log. Extensive Drag and Drop support. We could go on, but we just don't have the space here. So get a copy today, and start scripting with power!

+1 202 298 9595
ORDERS ONLY:
1 800 616 8320

www.mainevent.com
mainevent@his.com
AppleLink: MAIN.EVENT

1814 Belmont Road, NW
Washington, DC 20009 USA

Scripter is a registered trademark, and ScriptBase is a trademark, of Main Event Software, Inc. FaceSpan is a trademark of Software Designs Unlimited, Inc. Macintosh and AppleScript are registered trademarks, and Mac is a trademark, of Apple Computer, Inc.

17. **Don't share data structures between applications.** Although applications still run in the same address space, system services such as the Window Manager and Event Manager maintain their lists and queues on a per-application basis. You cannot create a data structure such as a `WindowRecord` in one application and pass it to another application.

18. **Don't hard-code font usage.** In Mac OS 8, users can customize the interface including application and system fonts. Use `GetSysFont` and `GetAppFont` instead of hard-coding fonts.

19. **Don't write to your application's data fork.** In Mac OS 8, your application's data fork is file-mapped into memory, and the operating system has exclusive read-only access to the file. Any attempts to access the file will result in a permission error. If you need to store data, store it in a resource or in a preferences file.

20. **Don't access hardware directly.** Under Mac OS 8, the hardware will be mapped into an address space which is not accessible by applications. Use drivers to access the hardware.

21. **Don't assume the system state in Notification Manager routines.** The times when Notification Manager completion routines are called in Mac OS 8 differ from when they are called in System 7. Specifically, don't rely on being in the context of the frontmost application.

22. **Don't change the window list directly.** In Mac OS 8, the system maintains the window list in a private manner. Don't assume that you can change the `nextWindow` field of a `WindowRecord` to change the window list. Use calls like `SendBehind` and `BringToFront` instead.

23. **Don't set the global-share bit in a CFM library that contains code.** Linking against per-context libraries in a globally shared library that contains code will not work in a per-context fashion. If you need to maintain system-wide global data, you should use a separate library which contains only data and doesn't link against other libraries.

24. **Don't rely on the structure of system memory.** In Mac OS 8, the relationships between the locations of application stacks, application heaps, code, and static data are private and should not be relied on. For example, don't assume your application code is in your application heap.

25. **Don't require AOCE interfaces.** Not all AOCE interfaces may be available in Mac OS 8. If your application uses AOCE, use `Gestalt` to check for the availability of AOCE features, and weak-link against the AOCE library.

26. **Don't require an INIT, control panel or desk accessory.** These code types are not supported in Mac OS 8; therefore you cannot rely on functionality provided by them.

27. **Don't use the Dictionary Manager.** The Dictionary Manager is not supported in Mac OS 8.

28. **Don't depend on Script Manager internals.** The Script Manager has been completely rewritten in Mac OS 8, and the internal data structures and low-memory usages have changed dramatically.

29. **Don't poll for Open Transport asynchronous events.** Endpoint providers should not use polling for asynchronous events. Use a notifier function in responding to asynchronous events.

30. **Don't pass open file reference numbers between processes.** In Mac OS 8, the FCB list is a per-process list, and the refNums cannot be shared among processes. Use file specifications or aliases instead.

31. **Don't assume that all FCB and VCB fields are valid.** The File Manager has been completely rewritten in Mac OS 8, and data structures like FCBs and VCBs are maintained for compatibility only. Only the absolutely required fields of these structures are maintained. The following FCB fields are maintained: `fcbFlNum`, `fcbFlags`, `fcbEOF`, `fcbPhLen`, `fcbCrPs`, `fcbVPtr`, `fcbClmpSize`, `fcbFType`, `fcbDirID`, and `FCBCName`. The following VCB fields are valid: `vcbSigWord`, `vcbVRefNum`, `vcbCrDate`, `vcbLsMod`, `vcbVolBkUp`, `vcbAtrb`, `vcbClpSiz`, `vcbNmAlBlks`, `vcbAlBlkSiz`, `vcbFreeBks`, `vcbVN`, `vcbDrvNum`, `vcbDRefNum`, `vcbNmFls`, `vcbNmRtDirs`, `vcbFilCnt`, `vcbDirCnt`, and `vcbFndrInfo`. Even though these fields are maintained in Mac OS 8, you should try to find alternate ways of gathering information provided by these fields.

32. **Don't modify FCB or VCB data structures.** As mentioned above, the FCBs and VCBs are maintained for compatibility only. The File Manager does not use these structures any more, so they need to be considered read-only.

33. **Don't call GetDrvQHdr or AddDrive.** In Mac OS 8, `GetDrvQHdr` points to a list which is always empty, and consequently `AddDrive` has no effect. Use specific File Manager calls to determine information about volumes.

### Steps to Being Well-Behaved in Mac OS 8

Once you have identified and addressed all applicable items discussed above, the next step is to start making some changes to your code to make it easier to build a Mac OS 8–savvy application.

# WHY NOT USE THE BEST?

**Companies like Adobe, Claris, Symantec, Metrowerks, Netscape and hundreds of others recognize the unsurpassed quality of Developer VISE and have switched to MindVision for their installation needs. In fact, every day MindVision sells more installers than all other competitors combined. Don't take our word for it. Ask any of our hundreds of satisfied customers why they switched from the competition to MindVision's Developer VISE.**

## We didn't stop there.
## Announcing...

# INSTALLER VISE 4.0

## The New Industry Standard for Software Installers.

**Find, move, rename, or delete any file. Create hierarchical packages. Build one installer for multiple languages. Build installers with unparalleled ease. Attach AppleScripts to your installation. New archive features allow you to build CD-ROM installers quickly and easily. It may not slice and dice, but in addition to installing it also removes. And much, much more.**

## And Introducing...

# UPDATER VISE 1.1

## The Most Advanced Updater Available.

**Updater VISE is the only product that can update multiple versions of your 68K, PowerPC, and Fat application from ONE updater. Updaters are extremely compact and totally secure, allowing for easy online distribution. And, combined with Installer VISE you can update any number of different files giving you a total software delivery solution.**

**Maybe it's time to stop and take a look at MindVision's software delivery solutions.**
**Visit our web site for full information,**
**including fully-functional demos.**

MindVision Software

# www.mindvision.com

Email: info@mindvision.com • Voice: (402) 477-3269 • Fax: (402) 477-1395

VISE SOFTWARE DELIVERY SYSTEM

© 1996 MindVision Software. All Rights Reserved. Developer VISE, Installer VISE, and Updater VISE are trademarks of MindVision Software.

1. **Use the latest Universal Headers.** All modern code bases should be using the Universal Headers instead of the old C includes or Pascal includes. Make sure you keep current with the latest releases of the Universal Headers. Products like MPW Pro, CodeWarrior, Symantec C++ all ship with the latest versions. By using the Universal Headers, you can help spot all the illegal references to low memory, use of unsupported traps, and reliance on unsupported system data structures – plus, you'll have the added benefit of using the same header files that will contain all the new Mac OS 8 functionality as conditionalized compiler switches.

2. **Make your code PowerPC-native.** While 68K applications are still fully supported in Mac OS 8, all new functionality is implemented through CFM and SOM, which means that your 68K code will not be able to take advantage of all the new features of Mac OS 8. Furthermore, performance tradeoffs in the design of Mac OS 8 all favor native PowerPC code rather than 68K code.

3. **Minimize patching.** In System 7, it was quite common for applications to patch traps to get a custom behavior with some part of the system, or to work around old bugs in the system. Many of these patches will not function correctly under Mac OS 8, and are not needed anyway. Check to see why you are patching any traps, and see if there is a better way to solve the problem.

4. **Support only System 7 and later.** Many applications contain old code which is still around for compatibility with System 6. It's time to remove that old code, and to set a new minimum system for your application. (Besides, it's easier on your QA department!)

5. **Use standard definition procedures.** In Mac OS 8, the user interface is greatly changed. Users can customize nearly all aspects of the appearance to their liking. These appearances are changed by using different window definitions, menu definitions, and control definitions. If you are using your own custom defprocs, the system cannot override them with the user-selected appearance, and your application will not have the same consistent look as other applications.

6. **Use Open Transport.** The networking implementation for Mac OS 8 is PowerPC-native Open Transport. The traditional AppleTalk and TCP APIs are still supported for backward compatibility, but performance is not optimal. For maximum networking performance, use native Open Transport calls.

7. **Use low-memory accessors.** When the first PowerMac models were introduced, new calls were added to provide access to supported low-memory locations. By using the `LMGet/LMSet` functions today, you can be assured that you aren't using any undocumented low-memory globals. In the future, low memory as we know it will go away completely, and the *only* way to access these locations will be through the `LMGet/LMSet` calls.

8. **Be virtual memory–friendly.** Virtual memory is always turned on in Mac OS 8, but the performance is improved as compared to System 7 VM. If you have been telling your users to turn off virtual memory when running your application, you should work to get your application running acceptably with System 7 VM. Make sure you are not making any System 7 VM calls like `LockMem` because they are not available in Mac OS 8.

9. **Locate special folders using the Folder Manager.** Use `FindFolder`, which has been implemented since System 7, to find the System Folder, Extensions folder, and other system-created folders. Store all user-defined preference data in the Preferences folder. Mac OS 8 implements a feature called "workspaces" which allows multiple users to define their own set of preferences for the same Macintosh. By storing your user preferences in the Preferences folder, your application will automatically take advantage of the workspaces in Mac OS 8.

10. **Factor your application.** Break your application into two distinct parts: the user interface part, and the computation part. If you can, create a third part which contains any disk-access and networking code. The benefits of this are many. **(1)** You will have an easier time producing a cross-platform version of your product. **(2)** It will be easier to add scriptability and recordability to your product. With Mac OS 8, the event model has changed from a polling model to a delivery model. [*Instead of having to ask over and over "Is there an event for me?", your designated routine is called if an event occurs; the event loop is dead in Mac OS 8. – man*] The changes you will want to make to your application to support this new event model will be very similar to the changes necessary to support AppleScript. **(3)** It will be easier to maintain both a System 7–specific and a Mac OS 8–specific version of your application. You can change only your interface code to take advantage of the new functionality in Mac OS 8, plus you can execute your disk-access and networking code as pre-emptive processes to achieve better throughput.

11. **Support QuickDraw GX printing.** Even though traditional printing calls are supported for compatibility in Mac OS 8, the preferred printing implementation is PowerPC-native QuickDraw GX. If you support the QuickDraw GX printing API, your application will print faster and more reliably. Support for both traditional and QuickDraw GX printing calls can be maintained by runtime Gestalt checks.

12. **Specify your stack size in the code fragment resource.** If your application needs stack space in excess of the standard allocated space, you should set the application

KILL YOUR BUGS BEFORE THEY KILL YOUR SCHEDULE!

TMON

Accelerated for Power Macintosh

PROFESSIONAL SYMBOLIC DEBUGGER

MindVision Software

www.mindvision.com
tel (402) 477-3269
fax (402) 477-1395

---

**StoneTable** Break on through the limitations of the List Manager

| | Name | Home Address |
|---|---|---|
| Father | **John Smith** | 2138 NE Halsey<br>Apt 3<br>Portland, OR 97232 |

Think C, CodeWarrior C, CodeWarrior Pascal, MPW C (68K or PPC)
Think Pascal, MPW Pascal, Prograph CPX (68K only)
68K StoneTable $150 StoneTableExtra $50 (per compiler)
PPC StoneTable $100 StoneTableExtra $25 (requires 68K)
International shipping (US Airmail) $10
No royalty fees for applications

CodeWarrior C package includes PowerPlant Class

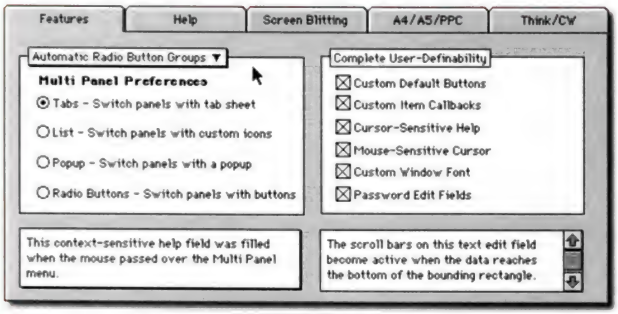
StoneTablet Publishing
P.O. Box 12665
Portland, OR 97212
voice/fax (503) 287-3424
stack@teleport.com

**StoneTable**
Row and column titles
variable size row and columns
move, copy, sort, hide, resize rows and columns
edit cells in place
validate data entry through callbacks
set font, size, fore/back color, face, alignment per cell
cell margins, top/bottom & left/right
68K support for both A4 and A5 worlds
"LDEF-like" custom drawing function
greater than 32K data per table
plus all List Manager functions and more

**StoneTableExtra**
drag cells in and between tables
PICT support
popup menus & check boxes
draw boxes around multiple cells
variable size grid lines
controls in cells

Demo - ftp://ftp.teleport.com/pub/vendors/stack/StoneTableDemo.hqx

GET AHEAD


Features
Help
Screen Blitting
A4/A5/PPC
Think/CW
Automatic Radio Button Groups
Multi Panel Preferences
Tabs - Switch panels with tab sheet
List - Switch panels with custom icons
Popup - Switch panels with a popup
Radio Buttons - Switch panels with buttons
Complete User-Definability
Custom Default Buttons
Custom Item Callbacks
Cursor-Sensitive Help
Mouse-Sensitive Cursor
Custom Window Font
Password Edit Fields
This context-sensitive help field was filled when the mouse passed over the Multi Panel menu.
The scroll bars on this text edit field become active when the data reaches the bottom of the bounding rectangle.

Meet Charlie. Charlie likes to program. But Charlie hates boring, repetitive code, so he created OpenDialog™. OpenDialog is "everything the Dialog Manager should've been." Use Charlie's head for dialogs and save your head (and your hair) for real problems. A set of libraries that replace the Dialog Manager, OpenDialog will

- Simplify creation of multi-panel items, such as preference dialogs.
- Relieve you of silly chores, like managing radio buttons.
- Provide you with "no-resource" dialogs for alerts & progress bars.
- Easily integrate into existing code.
- Cost only $259 (source code license available).
- Keep you from going bald. (Charlie lost his hair before OpenDialog.)

FGM Inc. • 703.478.9881 • http://www.fgm.com • email: opndlg42@fgm.com

Power Mac and Macintosh Developers:

FIND OUT FAST WHAT'S GOING ON IN MEMORY

THE MEMORY MINE™

- See memory allocation in any open heap at a glance.
- Easily spot memory leaks.
- Flags heap corruption when it happens.
- Works with source level debugger to let you find memory problems fast.
- Stress applications on the fly with Purge, Compact, and Zap.
- Allocate memory at will for precise stress testing.
- Log heap data - easily document heap status over time.
- No need for source code: nothing inserted in code; no patches to the system.
- Works with 24-bit, 32-bit, and modern memory managers.

For Macintoshes with 68020 or better. Requires System 7.0 or later.

only $99 US
Order now from Adianta, Inc.
Phone: (415)781-8052 • FAX: (415)781-8053
AppleLink:ADIANTA • AOL:Adianta • Internet:adianta@aol.com
For VISA, MC, or American Express orders by mail, fax, or Applelink, please include name, address, card number, expiration date, and phone number or email address.
Also available through the MacTech Mail Order Store and APDA

for more information contact

*Adianta, Inc.* • 582 Market St #911 • San Francisco, CA 94104


stack size field provided in the code fragment resource. In Mac OS 8, calls to `GetApplLimit` and `SetApplLimit` have no effect on PowerPC-native applications.

13. **Prepare for Mac OS 8 memory management.** Don't rely on any fields of the application heap data structures. Don't rely on any special knowledge of the behavior of the Memory Manager. Block headers, zone headers, and the size overhead of blocks are all private to the Memory Manager, and will change in Mac OS 8. Don't make assumptions about where and how memory blocks will move in the heap. Don't dispose of pointers and handles that are allocated indirectly by the system. (For example, don't dispose of the content region of a `WindowRecord`.) Don't access memory blocks outside your application heap.

14. **Use a supported framework.** Considering all the changes coming in Mac OS 8, now may be a good time to examine the application frameworks available. Using a supported framework will make it easier for you to take advantage of new functionality in Mac OS 8 by leveraging off the work the framework providers are doing to make their products Mac OS 8–savvy. Contact your compiler vendor to find out their plans for supporting Mac OS 8 with their frameworks.

15. **Be WorldScript-savvy.** Don't assume all text is in a single script. Don't assume a one-to-one relationship between characters displayed and glyphs; always use text-measuring routines on entire runs of text instead of character-by-character. Use `PixelToChar` and `CharToPixel` for hit-testing and highlighting. Use `GetFormatOrder` to properly reorder mixed-direction text.

16. **Be inline input–aware.** Follow the guidelines in *Inside Macintosh: Text*, Chapter 7, on what to do to become inline input–aware. This allows input methods to work directly in your document window without having to put up a separate window for text input. In Mac OS 8, inline input–aware applications will have access to a wide range of services besides input methods, such as spelling checkers and hyphenators.

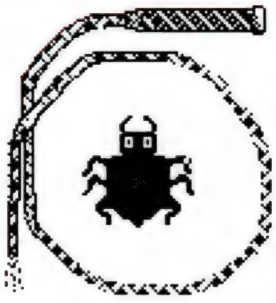
## SUMMARY

Mac OS 8 introduces some radical changes to Macintosh. Even so, Apple engineers have made backward compatibility a top priority. They have put a lot of work into making our lives easier; now it's time for us to return the favor. Make sure all your code follows the guidelines described above, and perhaps Mac OS 8 will ship a bit sooner than it would otherwise. So many times, we are faced with putting out a new version of a product without time to go back and clean up old code. By doing your cleaning *now*, you will not only be making your product better and more compatible with System 7, you'll be getting a head start in making your product work well with Mac OS 8.

**Gives you the Information to *Program your Best!***

# The Debugger V2 & MacNosy

*by Steve Jasik*



Information

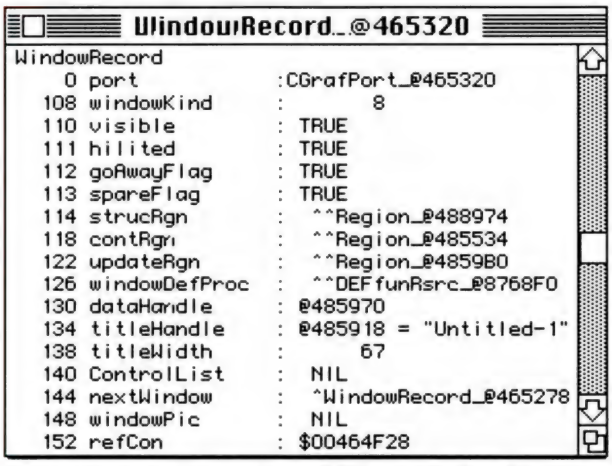
Control

For over 10 years Steve Jasik has brought Macintosh Developers the superior debugger. In addition to the software, Steve provides timely support to his customers. Inspect the feature list and see why so many developers use it. In addition to basic features such as stepping and breakpoints, the Jasik debugger has many advanced features such as heap display, a fast memory watch, Soft-MMU, jump tracing, ... to help you track down the elusive bugs that other debuggers won't help you find.

In additon, the package includes **MacNosy,** a global interactive disassembler that enables one to recover the source code of any Mac application or from the ROM.

When you compare features of the different debuggers, note that *only one* has all the below features to help you get your job done, and *only one* has MacNosy to help you debug any program in a full system environment symbolically!

"The Debugger" is the debugger of choice at: Adobe, Apple, Claris, Kodak, Metrowerks, etc.




*An example of a structured data display window with hypertext links to substructures*

## Its Features Include:

- **Source level debugging for Metrowerks & MPW** compiled programs (C++, C, Pascal, Fortran, ...).
- **Symbolic Debugging** of *any* Macintosh program, ROM, Plug-In or *code resource* (DRVRs, XCMDs, INITs, PDEFs..), ASLM & **CFM LIbraries, OpenDoc parts**, GX Drivers, ...
- **Simultaneous Symbolic debugging of multiple "tasks"**
- **Object Inspector for MacApp 3 & PowerPlant programs**
- **Source level debugging of Symantec C™ projects**
- **Includes a program (CoverTest) to interactively do Code Coverage analysis for SQA testing, etc.**
- **Fast Software Watchpoint** to find clobbered variables.
- **Soft MMU** ("heap centric" bounds checking) for PowerPC programs to find out of bounds references.
- **Jump Trace** to find where a program takes a "wild jump"
- **Sophisticated error check algorithms** such as Trap Discipline (Argument Checking), Heap Scramble and Heap Check to detect errors before they become disasters
- **Structured display** of data (hypertext) with user definable structures while debugging
- **Conditional breakpoints** to filter redundant information
- **Continuous Animated Step Mode** to watch your program execute instruction by instruction
- **Detailed symbolic disassembly for both 680x0 and PowerPC** with symbol names, labels, cross ref maps, – make it possible to ferret out the secrets of the ROM, etc.
- **"Training Wheels"** for the PowerPC disassembler to help you learn the opcodes

**The Debugger V2 & MacNosy: $350**

Runs on all Macs. Call For Group prices or Updates. Visa/MC Accepted.

**Available from:** Jasik, APDA or ComputerWare (800-326-0092).

**Jasik Designs • 343 Trenton Way, Menlo Park, California 94025 • (415) 322-1386**

**URL: http://www.jasik.com e-mail: macnosy@jasik.com**

# UNIFORM RESOURCE LOCATORS

*By Jim Straus, URLs@mactech.com*

Don't hesitate to email me with any new sites! The full list is always available on-line at http://www.mactech.com/URLs.html.

## QUICKIES

### Internet-Related

| | |
|---|---|
| Today's Browser Idea | http://www.smartbrowser.com/idea/index.html |
| Tell It Discussion Forum | http://cmp.goshen.net/tellit/ |

### Vendors, Products and Miscellaneous

| | |
|---|---|
| MacSense Shareware | http://www.macsense.com/MacSense/Resources/Shareware.html |
| Miracle Software | http://webster.northnet.org/ |
| Vietnamese for Mac | http://www.webcom.com/hcgvn/software/mac/mac.html |
| Water's Edge Software | http://www.interlog.com/~wateredg |

## WEB WATCH

### Internet

The rapid advance in Internet browsers is amazing. Microsoft and Netscape are fighting over your disk space and each is introducing new versions of their software at a breakneck pace. You owe it to yourself to make sure you have the latest of whichever browser you prefer (or do as I do and have both). To keep abreast of what is happening with browsers, there is the very useful BrowserWatch site. Here you can find all the rumors of upcoming releases and information on plug-ins, bugs, and statistics on which browsers are the most popular.

**Netscape Navigator**
**http://home.netscape.com/comprod/mirror/index.html**

**Microsoft Internet Explorer**
**http://www.microsoft.com/ie/platform/macmsie.htm**

**BrowserWatch**
**http://www.ski.mskcc.org/browserwatch/**

To go along with the new capabilities of the browsers, the Web page creation and editing tools are having to chase to keep up. Besides Adobe's PageMill, there is the World Wide Web Weaver and PageSpinner. I've also heard about an editor called Loma Prieta; maybe it will be available when you read this. And there is always BBEdit with the HTML extensions.

**Adobe PageMill**
**http://www.adobe.com/prodindex/pagemill/main.html**

**World Wide Web Weaver**
**http://webster.northnet.org/Web.Weaver/WWWW.html**

**PageSpinner**
**http://www.algonet.se/~optima/pagespinner.html**

**BBEdit**
**http://www.barebones.com/**

### Macintosh

ScriptWeb has been updated and has more script system information than ever. This comprehensive source of information about scripting on the Macintosh is invaluable to anyone doing any script development.

**ScriptWeb**
**http://www.scriptweb.com/**

Apple's Developer University is making its courses available on the Internet. Based on its self-paced courses and updated and hypertext-linked together, these will be a great resource for developers. Even if you're an experienced Mac developer, sometimes you have to delve into new areas. The Mac system software is getting so big that it is difficult to keep up with everything. The first sets of courses available are Drag and Drop, Thread Manager, QuickDraw GX, and PowerTalk.

**Apple Developer U. On-line**
**http://dev.info.apple.com/du/7.5online/7.5TopicsText/ST01-7.5Intro.html**

Ambrosia is hosting a large number of sites interesting to Mac aficionados and developers. Available here is information on Mac Game resources, MacMedia magazine for users, the alt.sources.mac archive, the Mac Internet Page, the PowerMac Pruning Page, and others. They don't create all this content, but are hosting a lot of good stuff.

**Ambrosia Cafe**
**http://www.AmbrosiaSW.com/Cafe.html**

### Neat Non-Macintosh Site

If you never read it, you can now. *Engines of Creation* by K. Eric Drexler is now available on the Web. This seminal book talks about how nanotechnology is going to radically change the world. If you have not heard about nanotechnology, you probably should. It is likely to become part of the political and scientific debates in the future. Even if you have read the book, it might be worth it to read it again, and the on-line version has links to sources of further information.

**Engines of Creation**
**http://reality.sgi.com/whitaker/EnginesOfCreation/**

I originally mentioned this site back in one of my first columns, as having a page for looking at fractally generated planets. Besides that, there are various code sections that might be useful to a developer (such as a string macro language, the Hacker's Diet, and code to compress voice to 2400 bps). There are also speculations on semi-autonomous mining robots, a "final chapter" to Rudy Rucker's novel *The Hacker and the Ants*, and a 900-page history of Autodesk.

**Index Librorum Liberorum**
**http://www.fourmilab.ch/**

Wow! Oh! Yeah! Oo! Right On!

# Recent Reactions To New QUED/M 3.0

Why does the new QUED/M 3.0 text editor elicit such responses? Because it's so fun to use, that's why. From the moment you see its inviting interface and access its numerous time-saving features, you'll likely have a few happy responses of your own to give.

QUED/M 3.0 makes editing fun by making it easy. Some of the many features you'll appreciate right away:

- Macros with a powerful new Programming Dialect provide built-in, Basic-like commands
- Integrated support for THINK C, CodeWarrior, MPW 411 and ToolBox Assistant lets you use QUED/M's advanced editor for all your editing

- Dynamic coloring of C/C++ keywords and comments color codes your work
- Unsurpassed editing capabilities, including System 7.5 Drag and Drop, noncontiguous selection, unlimited undos, GREP find/replace, Frontier™ and AppleScript support, file comparison, text folding, and more!
- Power PC code speeds up editing, searching and macros
- HTML macros included for creating Web pages
- FREE Apprentice 4 gives you over 600 MB of working source code to include in your own programs. MailKeeper e-mail organizer is included too!
- All this and more, for less than you'd pay for other text editors!

**Start Having Fun. Order Qued/M 3.0 Today.** Get your copy for just $69 at CYBERIAN Outpost **800-856-9800,** http://www.cybout.com/ • info@cybout.com. Or for more information, call 800-281-0101, or 619-481-4366 for Info-by-Fax. http://www.nisus-soft.com

NISUS Software Inc.

©1996 Nisus Software Inc. QUED/M is a trademark, and Nisus is a registered trademark of Nisus Software Inc. All other product names are the property of their respective owners. Nisus main line: 619-481-1477

---

## TCP/IP Scripting Addition — *The Internet Scripting Solution*

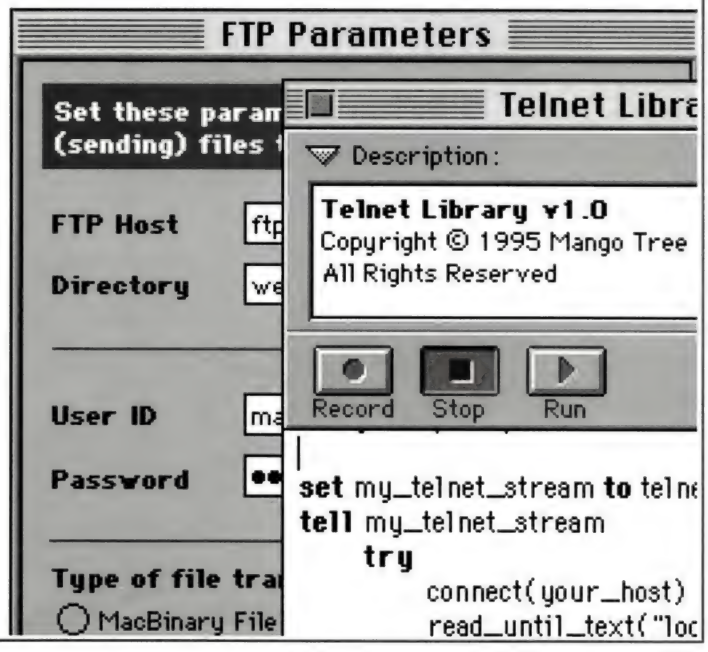
The **TCP/IP Scripting Addition** allows you to quickly develop Internet client/server applications using AppleScript®. If you want to script with MacTCP ™ and Open Transport™, here's your solution!

- Supports ScriptEditor, FaceSpan ™, and HyperCard™
- Build net–wise WebSTAR ™ CGI scripts and NetScape™ CCI scripts
- Sample scripts include FTP, Gopher, Telnet, Post Office, E–Mail and more
- Featured on the Apple® Internet Server

ONLY $49 SRP

**Order now through the MacTech Mail Order Store at 805–494–9797 or other mail order stores**

**Mango Tree Software, Inc.**
Box 1057 • Brookline, Massachusetts 02146
617–327–8663
http://www.mangotree.com/biz/mango




All trademarks are properties of their respective holders.
Contact Mango Tree Software for site licensing and redistribution information.
Copyright © 1996 Mango Tree Software, Inc.

# SoftPolish

## The Essential Tool for Software Quality

### Introducing SoftPolish

SoftPolish is a quality assurance tool that helps you deliver applications that are clean, consistent, and free from spelling and user-interface mistakes.

### Clean 'til it Shines

Apply SoftPolish to your product's resource file (or built application). SoftPolish checks file integrity and scans the application signature, menu and 'cmnu' resources, controls, dialogs, pictures, strings, 'TEXT' resources, windows, and Balloon Help™ resources. It reports any structural, spelling or typographical errors that it encounters. SoftPolish also checks the layout of dialogs and controls to make sure that items don't overlap. At your request, SoftPolish will also correct spelling mistakes, so you'll never again ship a product with embarrassing typos.

### Wipe Up Old Spills

When applied to a 68K application, SoftPolish also catches code that you wouldn't want your customers to see: `Debugger` and `DebugStr` calls, `CHK` and `TRAPV` exceptions, and MacsBug symbol names.

### The Finishing Touch

When preparing your product for release, use SoftPolish to ensure a perfect presentation to the world: it cleans up unwanted resources in your files (such as extraneous 'SIZE' and 'ckid' resources), sets creation and modification dates to a consistent value, and checks the spelling of file and folder names. After all, your customers don't really need to know that the product was built at 4:30 AM on the day after you announced... Do they?

http://www.barebones.com
or
617•676•0650

SoftPolish is a trademark of Bare Bones Software, Inc. All other trademarks and registered trademarks are properties of their respective holders.
© 1996 Bare Bones Software, Inc.

*By Dave Mark*

# Heidi Ho!

*Dave was offered a chance this month to interview Heidi Roizen, Apple's Vice President for Developer Relations, and we felt that, even though it's a departure from his usual "Factory Floor" beat, the opportunity to let her share her thoughts with our readers was too good to pass up. Let us know what you think; perhaps we'll do more of this sort of thing. – man*

**Dave:** Tell me about your organization.

**Heidi:** The role of Apple Developer Relations (ADR) is to try to ensure that there is a rich business and technical proposition for supporting our platform, as well as to be the principal touchpoint between the developer community and Apple. ADR is made up of five groups:

- The Evangelism group (Brian Gentile, gentile@applelink.apple.com) is responsible for the day-to-day account management of developers with whom we need to have a one-on-one relationship at any given time. The group can handle about 500 companies, those of the traditional "volume" developer variety as well as the smaller companies in emerging markets or key customer segments. In addition, the group is responsible for technology evangelism and adoption by developers, including activities such as briefings, seeding, and developer kitchens. Most importantly, this activity also involves bringing the developer message about our technologies back to Apple inside folks, who can then develop our strategies and technologies in concert with developer adoption and exploitation.
- The Developer Marketing group (Jonathan Fader, fader@applelink.apple.com) handles the marketing side, with co-marketing programs, market research, services and general account management of the 11,000+ companies which have registered with Apple as developers. Developer Marketing Programs include Apple's membership-based Developer Programs, the Apple Developer Catalog, developer communications and periodicals (such as *Apple Directions* and *develop*), and the technical marketing of software technologies, tools, and training products. Further, the group is now expanding its role to help developers market to customers, with general programs aimed at the channel, the internet, and more targeted co-marketing programs with Apple's sales groups, both domestically and globally.
- The Technology Services group (Garry Hornbuckle, garryh@apple.com) manages the technical side of the equation. This group includes Developer Technical Support, Developer technical information (Web and online content, hardware and software documentation), Developer University content, compatibility and testing labs, and specialized engineering support for developers. This group is also expanding to come up with new ways to meet developer needs with highly leveraged investments in support, technology, training and tools.

***Heidi Roizen*** (heidi@apple.com) is an incredibly nice person who happens to be the VP of Apple Developer Relations. Heidi is responsible for building and maintaining relationships between Apple and its developers. Before she moved to Apple, Heidi was co-founder and CEO of T/Maker, one of the very first Macintosh software companies. Heidi was a member of the board of directors of the Software Publishers Association, and served as its president from 1988 to 1990. Currently, she serves on the advisory board of the Software Forum, the largest independent development organization in the country. She is also a member of the Stanford Alumni Association board of directors, and of the board of governers of the Pacific Stock Exchange.

- The Business Development group (Heidi is covering this until someone permanent is hired) covers licensing and other fundamental business transactions. This group is also responsible for understanding the business needs of Apple's key developers and for understanding the implications to developers of Apple's strategic directions.

- Finally, the International Developer Relations group (David Krathwohl, krathwohl@applelink.apple.com) provides developer relations support outside the US and Canada, in all the above areas of specialty.

  With over 11,000 companies in our portfolio of responsibility, we've tried to organize ourselves in a way that gets to real action and real progress with the fewest steps for both you and Apple, and that systematizes things so we're not all just reinventing others' wheels. Admittedly, we're still in the building phase in every one of these areas, so we don't always have the necessary pieces in place, but we're working on it. We're trying to devise real programs that can be offered to many developers to achieve the highest leverage for the dollars we spend.

  I know that in the past Apple has seemed to waver on its commitment to working with the developer community, alternately looking upon it as a giant democracy and as a profit center. I believe that in order to do the maximum good for the platform, we must recognize that Apple's commitment to working with developers is a cost center, but a highly, highly leveraged cost center. Many small companies approach me saying that since we're "great big Apple", we can afford to do a lot for them. I have to remind everyone that even you "small guys" probably made about $700 million of profit more than we did in the last quarter, so our resources need to be very carefully managed. Having said that, our new CEO, Gil Amelio, has committed significant resources to providing technical and business support for Apple developers in a highly leveraged, "obvious value" way, leveraging Apple's dollars to benefit the platform, and leveraging your dollars on top of that by using our collective bargaining power, channel presence, or whatever we have to bring to the party.

**Dave:** At WWDC, Gil Amelio announced that Apple had allocated $20 million for developer co-marketing. What can you tell us about that?

**Heidi:** As you said, Gil Amelio has allocated $20 million over the next 12 months, to help our developers get their products in front of customers. The programs that result will fall mostly under the Developer Marketing group headed by Jonathan Fader, and the International group headed by David Krathwohl.

  The money will be spent roughly half in the US and Canada and half in Apple's markets outside the US. We are working on programs now that will best leverage this money in both the traditional channel as well as in some emerging channels. The funding is likely to be split among channel initiatives, "virtual" and Internet initiatives, comarketing and collateral materials, and some end-customer advertising to tell Mac users what is available for them and where to buy.

  In order to maximize the leverage, we aren't going to be looking at specific proposals from our developers, but rather will take their input and create a few standard programs that can either be shared by all, be "pay to play", or be targeted to particular segments of the market. I would certainly encourage developers to give ideas to Jonathan's team, and we'll keep them in the loop as the programs become reality, but I'm afraid that a specific proposal unique to any one developer is outside the bounds of what the money was for.

**Dave:** What changes do you see in the relationship Apple has with its developers?

**Heidi:** We need to treat our developers as the important business partners they are. We need to recognize their needs as businesses, including the issues they are dealing with in the channel, the rising cost of development, the need for many to be cross-platform, and much, much more.

  We also need to build person-to-person relationships with many of the developers on whom our future customers will depend. We need to provide both marketing and technical support in a consistent, programmatic manner. We need to give them world-class tools and technologies that enable them to solve customer needs on Apple platforms like nowhere else. In short, there's no shortage of things for me to do in my new job...!

**Dave:** It used to be that developing for the Mac was more profitable than developing for Windows. Now, Windows tools and developer programs have improved greatly (some would say they've become better than Apple's), while developing for the Mac has become much more complex. How will you solve this problem? How will you make it more profitable to take Mac applications to market? How will you make it more attractive to develop for Mac first instead of the more normal Windows first?

**Heidi:** Wow, that's a mouthful! First of all, developers are simple (at least I was when I was one): developers are first and foremost businesses. They need to meet customer needs. They need to make a profit. To make a profit they need to spend less money than they make. I believe Apple can positively influence all those points. We can streamline

the product line and improve our testing resources so that it is easier to develop for and certify on our platform. We can improve the revenue (volume) side of the equation through licensing. We can reduce costs through breakthrough, enabling technologies that greatly simplify the cost and time-to-market of creating competitive products. Those are tall orders. Perhaps, more simply, we can help developers get to more customers, so that more units are sold and more profit is made on our platform. There are plenty of developers on all popular platforms today making money; there are also plenty of developers on all popular platforms not making money.

Everything is changing for the developer today, from the development environment, to the customer, to the channel, to the type of product desired, even to the definition of the platform. The more we can stay on top of these changes and help move them into the direction of the greatest economic good for our developers, the more developers we'll make happy. Remember this: there was over $1 billion in Mac software sold last year, so there's still a very healthy market out there today. But, we can do better, I believe.

**Dave:** How will you address the shelf space problem (the lack of computer store shelf space for Mac applications)?

**Heidi:** I can give you two answers to this: either leverage with a very few channel partners to literally open up shelf space and drive Mac customers to that space; or, if we don't believe that will work, take the next step in where the channel is going (i.e., the Internet) and be there more aggressively and progressively than anyone else. Lead the charge, if you will. I'm not sure what we're going to do yet, as we're just researching the options right now, but we're targeting having some answers very soon.

**Dave:** What will Apple do to make life easier for small developers?

**Heidi:** Some of the things I'm going to say will sound very basic, but guess what – they are! It's nice that some answers are so simple.

First of all, we're going to make it easier and more immediate to get information about Apple – our technologies, markets, distributors, who at Apple does what, how do bundle programs work, what will ship when, and multitudes of other business and technical information that will make it incredibly easier to work with Apple as a business partner. We're going to paper as a last resort, using the Web, CDs and email as our principal methods of information dissemination. Beyond that, we're going to try to use our leverage to create better marketing programs and technical programs where, even if a developer is asked to defray some portion of the cost, it will be a no-brainer in terms of value or benefit. As a former Mac developer for over a decade, nothing used to burn me more than being asked to pay for something which was ultimately a bad use of my money. I'm going to do my best to make sure that doesn't happen in our group, or at Apple at large.

**Dave:** Apple has a lot of different technologies on its plate. Traditionally, programmers implement a small subset of these technologies, while some third parties try to find holes in Apple's strategy that they can fill with their own solutions (StuffIt, AppleScript debuggers, etc.). Apple keeps the technology development within, while third parties get the crumbs. Is Apple going to take some of these technologies off its own plate and allow third parties to take over some of these projects?

**Heidi:** This is a huge issue for Apple. We should not gratuitously compete with our own developers in our own markets. Having said that, we also have to compete with someone, have to add unique value to our platform and to our products, and earn our own profits, or our future will be pretty grim. I believe that as a first step, we need to understand what our customers need. Next, we have to figure out the best way to deliver that, through our own product creation efforts as well as through working with others. My own bias is that we'll deliver much better product much more efficiently by working with outside parties; after all, it effectively doubles our R&D dollars. Having said that, every case will be unique and needs to be thoroughly thought out, negotiated, determined. I'd say the positive news in all this is a fundamental shift in Apple's philosophy to work outside first, as opposed to the other way around. Still, it has to make business sense for Apple too, or the whole picture ultimately falls apart.

**To receive information on any products advertised in this issue, send your request via Internet: productinfo@xplain.com**

**Visit MacTech Magazine's Web site!**
http://www.mactech.com

# INTRANET RAPID APPLICATION DEVELOPMENT

Tango

# Instantly connect your Databases to the World-Wide Web. Any Database! Any Platform!

## TANGO FEATURES

- Graphical development environment used to create dynamic Web-based applications
- Automatically generates the HTML and SQL for you
- Allows access to FileMaker Pro and ODBC compatible databases
- Supports global variables allowing for state persistence
- Allows other applications to be called through Apple Events
- Support for executing custom SQL or AppleScripts

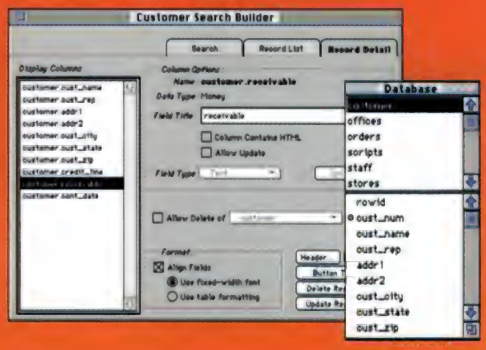
## CROSS PLATFORM CAPABILITIES

- Develop on Macintosh or Windows
- Deploy on any Web server running on Macintosh, Solaris or Windows NT
- Access any database through ODBC

## SUPPORTED DATABASES

- FileMaker Pro
- Oracle
- Sybase
- Microsoft SQL Server
- FoxPro
- Butler SQL
- Access

Tango's built-in wizards called Builders, allow you to easily create Query Documents in a drag-and-drop environment




The Query Document defines the actions within the Web application–from database searches to logical flow control

Product Catalogs like these are created with just a few clicks of the mouse

Tango— an Intranet Rapid Application Development tool (IRAD) that allows Webmasters to quickly connect their databases to Web servers through a visual drag-and-drop development environment.

If you want to have your database on-line quickly, without writing complex code, visit EveryWares Web site at: **http://www.everyware.com** and receive an introductory offer on *Tango for FileMaker Pro* or call toll free: **1-888-819-2500**

EVERYWARE

© EveryWare Development Corp. 1996. Tango is a trademark of EveryWare Development Corp. All other products mentioned are trademarks of their respective companies.

Tango IRAD

*By Lawrence D'Oliveiro*

# Dragging and Rigid-Body Transformations

***QD GX can make your Mac a swinging place***

QuickDraw GX offers some interesting new possibilities for interactive graphics. This article is a note on one of them: the idea of dragging an object about, and having it rotate to follow the drag, just like real objects tend to do when you pull or push them.

The code I'm presenting here is by no means finished; think of it as a rough sketch, a demonstration of the concept. It is written in Modula-2, and makes heavy use of my standard libraries, which you can find at ftp://ftphost.waikato.ac.nz/pub/ldo/MyModLib.hqx.

Feel free to use my code as a starting point for your own experiments.

## Transformations in QuickDraw GX

QuickDraw GX supports 3-by-3 **mappings**, which are capable of applying any kind of **linear transformation** to the geometry of a GX shape. (A linear transformation is one that preserves the straightness of lines. Thus, a GX mapping cannot transform a straight line to a curve, or vice versa.)

Linear transformations can be broken down into various simple types: translations (changes of position), rotations, scaling, skewing and perspective. In fact, any arbitrary linear transformation can be considered to be built from components of these types.

There are various useful subsets of linear transformations: an **affine transformation** is one where any pair of parallel lines remains parallel after the transformation. GX's perspective transformations clearly are not affine; thus, an affine transformation is one that has no perspective component.

An important subset of affine transformations is the set of **rigid-body transformations**: these are ones that preserve the distance between any two given points. Rigid-body transformations consist only of translations and rotations; this corresponds intuitively to our notions of real rigid bodies, which cannot be scaled or stretched in any way, though you can usually move them around and reorient them any way you like (unless they're a lot bigger or heavier than you are).

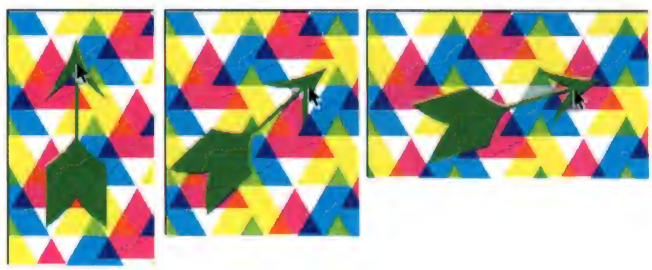


*Figure 1. As the cursor pulls the arrow to the right, the arrow swings into line with the direction of motion. These are successive screen shots from my program; the upper-left corner of each shot is the upper-left corner of the window.*

## Dragging a Rigid Body

Consider what happens when you try to move a real rigid body. The body has a center of mass; if you orient the direction of your force so that it passes exactly through this point, the object will change its position without rotating. But if you offset your

***Lawrence D'Oliveiro*** delights in taking unusual, even perverse approaches to common problems. He can be reached on AppleLink at ldo@waikato.ac.nz@internet#, and on the Internet at nz0027@applelink.apple.com.

force by any amount, you will produce a torque, and the object will change both its position and its orientation. (For example, place a book on a smooth surface such a desk, grasp it by just one corner, and pull in a straight line; the book, as it starts to move, also swings around to line up its center of mass with the direction of travel.)

Figure 1 shows this happening in my program.

In the following analysis, I'll follow the rules of Aristotelian, rather than Newtonian, physics: objects have no inertia, but they are subject to friction, which acts through their center of mass. Thus, they stop moving as soon as you stop pushing them.

Consider a body with its center of mass at a point **C**. Say you apply a small force on it at a point **F**, sufficient to displace that point to a new position **G** close to **F**. The center of mass is in turn moved to a new position **D** (Figure 2).

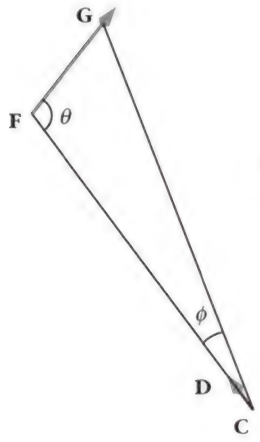



*Figure 2. Movements...*

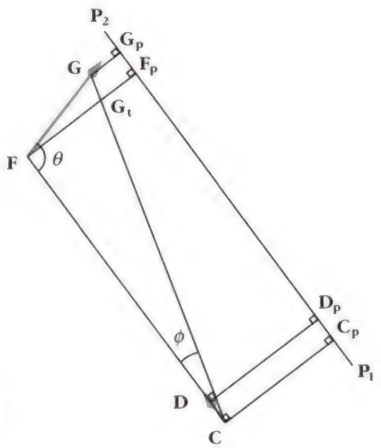



*Figure 3. ...and analysis*

The force you apply can be split into two components: the component parallel to **CF** moves the object without rotating it (hence the new position of the center of mass, **D**, lies on this line), while the component perpendicular to this line exerts a pure rotational force on the object without moving the center of mass.

In Figure 2, θ is the angle **CFG** and can have any value, while ϕ is the angle of rotation **FCG**, and is assumed to be small (you'll see why shortly).

Now draw a line $\mathbf{P_1P_2}$ parallel to **CF**, and project the points **C**, **D**, **F** and **G** onto this line at the points $\mathbf{C_p}$, $\mathbf{D_p}$, $\mathbf{F_p}$ and $\mathbf{G_p}$ (Figure 3). $\mathbf{G_t}$ is the intersection between $\mathbf{FF_p}$ and **CG**. In this case, θ is greater than 90°, so the angle $\mathbf{GFG_t}$ is θ - 90° (I'll leave the analysis of the case where θ is less than 90° as an "exercise for the reader", as they say). Since ϕ is small, $\mathbf{FG_tG}$ is close to 90°, so the ratio $\frac{\mathbf{FG_t}}{\mathbf{FG}}$ is approximately cos θ - 90°, which equals sin θ. Now, since $\mathbf{GFG_t}$ is a right angle, sin ϕ equals $\frac{\mathbf{FG_t}}{\mathbf{CF}}$ which becomes $\frac{\mathbf{FG}\sin\theta}{\mathbf{CF}}$ when you apply the above approximation.

For larger movements, where the angle ϕ might not be small, simply split the movement into lots of small steps with correspondingly small ϕ. If you want a fancy-sounding term for this mathematical trick, it's called differential calculus.

### Translating This Into Code

The next step is to write some actual code based on this analysis. In the following, I'll intermingle declarations and statements to suit the exposition, rather than the requirements of strict language syntax (in other words, I'll be following the order in which code is usually written).

```
VAR
  LastMouse, ThisMouse : Point;
  ThisDelta : gxPoint;
  ShapeCenter : gxPoint;
```

`LastMouse` is the point **F**, while `ThisMouse` is the point **G**. `ThisDelta` is the displacement **FG**, computed as follows:

```
ThisDelta.x := IntToFixed(ThisMouse.h - LastMouse.h);
ThisDelta.y := IntToFixed(ThisMouse.v - LastMouse.v);
```

`ShapeCenter` is supposed to be the center of mass, or center of geometry, of the shape. GX provides a call, `GXGetShapeCenter`, that is supposed to return this in the coordinate system of the shape geometry itself. You could then put this through the shape mapping to get the center in "local coordinates". Unfortunately, when I tried this, I got incorrect results for a complex picture shape (a QuickDraw GX bug?). So, to avoid this problem, my code takes `ShapeCenter` as the center of the shape's bounding rectangle, rather than of its actual geometry:

```
GXGetShapeLocalBounds(TheShape, Bounds);
ShapeCenter.x := (Bounds.right + Bounds.left) DIV 2;
ShapeCenter.y := (Bounds.bottom + Bounds.top) DIV 2;
```

Next we perform the computation of the actual rotation angle, using good old floating-point numbers instead of fixed-point ones:

```
VAR
  DeltaX, DeltaY, Delta : LongReal;
  RadiusX, RadiusY, Radius : LongReal;
  DragAngle, RotationAngleSin : LongReal;
  RotationAngle : LongReal;
```

`DeltaX` and `DeltaY` are the x- and y-components of the displacement **FG**, while `Delta` is the magnitude of this displacement.

```
DeltaX := Fix2Double(ThisDelta.x);
DeltaY := Fix2Double(ThisDelta.y);
Delta := Sqrt(Squared(DeltaX) + Squared(DeltaY));
```

Similarly, `RadiusX` and `RadiusY` are the x- and y-components of thedistance **CF**, while `Radius` is the magnitude of this distance.

```
RadiusX := Fix2Double(IntToFixed(LastMouse.h) - ShapeCenter.x);
RadiusY := Fix2Double(IntToFixed(LastMouse.v) - ShapeCenter.y);
Radius := Sqrt(Squared(RadiusX) + Squared(RadiusY));
```

`DragAngle` is the angle θ. I compute it here from the

difference between the angles of the lines **FG** and **CF** (`ArcTan2(x, y)` returns the angle with tangent `y/x` in the appropriate quadrant, taking account of the signs of `x` and `y`):

```
DragAngle := ArcTan2(DeltaX, DeltaY) - ArcTan2(RadiusX, RadiusY);
```

`RotationAngleSin` is the sine of the angle $\phi$. `DLimit` is just a routine that constrains its first argument to within the specified limits (in this case, between -1 and 1). The need for this constraint will become apparent later.

```
RotationAngleSin := DLimit
  (
  Delta * Sin(DragAngle) / Radius,
  FLOATD(-1),
  FLOATD(1)
  );
```

And finally, we compute the angle $\phi$ itself, converting from the floating-point radians that SANE operates in, to the fixed-point degrees that QuickDraw GX uses:

```
RotationAngle :=
  Double2Fix(ArcSin(RotationAngleSin) * FLOATD(180) / Pi())
```

Applying the transformation to the shape is pretty straightforward. Simply obtain the shape's existing mapping:

```
VAR
  ShapeMapping : gxMapping;
...
GXGetShapeMapping(TheShape, ShapeMapping);
```

then apply the appropriate movement and rotation:

```
MoveMapping
  (
  (*@target :=*) ShapeMapping,
  (*hOffset :=*) ThisDelta.x,
  (*vOffset :=*) ThisDelta.y
  );
RotateMapping
  (
  (*@target :=*) ShapeMapping,
  (*angle :=*) RotationAngle,
  (*xCenter :=*) IntToFixed(ThisMouse.h),
  (*yCenter :=*) IntToFixed(ThisMouse.v)
  );
GXSetShapeMapping(TheShape, ShapeMapping)
```

Note that the center of rotation used in the `RotateMapping` call is the point **G** rather than **C**. This is all right, because the direction of rotation calculated by the code is actually the opposite of that in the analysis; thus the effect is the same.

### Other Matters

There are several other aspects of the example code that I've glossed over so far.

To keep the size of the source code down, the program itself has absolutely the minimum user interface I felt I could get away with. It doesn't even have any menus! When you start it up, it already has a shape loaded that you can try dragging about. You can bring in a different shape by dragging it into the window, from a Finder clipping file or another drag-aware application (I've provided a few sample clippings you can try). To quit the program, click the close box in the window.

An important issue is how to do off-screen rendering, so that your on-screen drawing doesn't flicker. There is no direct equivalent to QuickDraw GWorlds in QuickDraw GX, but the GX SDK library code shows you how to create a much more powerful alternative: an offscreen graphics context that can optimize drawing simultaneously for multiple screens, rather than just the deepest one that your window happens to cross.

My actual program uses a routine I wrote called `MoveSprite`, which creates temporary offscreen structures every time you call it to move an object. It automatically sizes these structures to cover only the affected on-screen area (and doesn't bother doing off-screen drawing if the old and new positions of the object don't overlap). Thus, this routine is simpler to use than explicitly creating separate off-screen structures and reusing them for the duration of the drag, though it may be slower.

One feature of the code is that it doesn't rotate the shape if you drag it at a point close to its center, or if you hold the command-key down. I figure this sort of feature could be useful in a "real" program, where the user might not always want the object to rotate.

One pitfall you always have to keep in mind when doing calculations on a computer is rounding error. In this case, repeated calls to `RotateMapping` can accumulate distortions in the shape geometry, since the mapping elements cannot be computed exactly. To get around this, you should keep separate track of the shape position and rotation angle, and recompute the shape mapping from these values each time, instead of applying an incremental rotation to the previous mapping. This will keep the distortion within fixed bounds, instead of letting it accumulate. My code doesn't do this, though I must admit I have yet to notice any distortions in shapes after repeated drags.

Finally, I should own up to one important liberty I have taken with the mathematical analysis. Remember how I kept saying that the displacement and rotation angle were assumed to be small? In fact, the code will happily compute arbitrarily large values for the **FG** displacement, depending on how quickly you can move the mouse, and how long it takes your machine to redraw the shape in between checking the mouse position. This means that, for large mouse movements, the code is no longer strictly MC (Mathematically Correct). It's also why I put in the `DLimit` call.

But then, I figure this is part of the fun of programming. For instance, I have found that I can drag a shape to a corner of the window, and leave the mouse absolutely stationary outside the window, while the shape continues to spin round and round in the corner. Is this a consequence of the code hitting a non-MC situation? I don't know – you tell me!

*By Jeremy Vineyard*

# Debugging with QC

## *Stress your software, not yourself*

### Introduction

The Memory Manager is one of the most commonly used components of the Mac Toolbox. Most applications are continually allocating and releasing memory – loading windows, pictures, and other resources, locking them, relocating them, resizing them, and purging them when they have served their purpose.

With all of this activity in memory, it is easy for a programmer in a hurry to overlook sections of code that could cause unforeseen problems for users. Attitudes like, "This memory operation will never fail", and, "It's only a small memory leak", frequently lead to software that crashes, frustrating the user to the point of switching to your competitor and advising associates not to buy your product.

Because the Memory Manager is so crucial to any Mac application, it is crucial to ensure that the use of the Memory Manager is solid. The problem is that, most of the time, the effects of memory-related bugs are seen long after the bug occurs, making tracking them down a difficult and time-consuming task.

Thankfully, there is a tool available that makes preventing and tracking down hard-to-reproduce memory-related bugs easier. This is QC (as in "Quality Control"), a low-level debugging tool for use with the Mac OS Memory Manager.

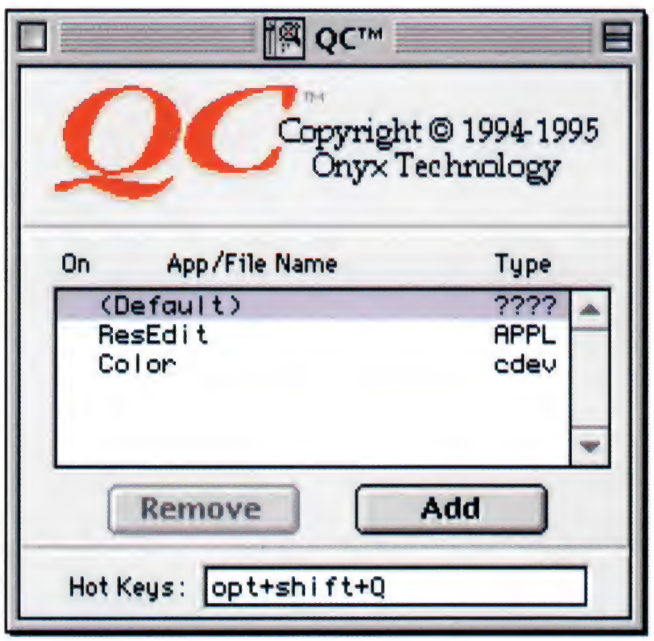



*Figure 1. QC Control Panel*

### Setting Up and Using QC

QC requires System 7 in order to run. It is developed by Onyx Technology and costs around $100. QC is copy-protected and will allow you to run only one serialized copy at a time on a network. Additional serial numbers are available by buying additional copies of QC.

QC consists of a control panel, on-line documentation, and a set of C APIs. All interaction with QC is done in the control panel.

***Jeremy Vineyard*** works for Farallon Computing, Inc., and is a software engineer working on Timbuktu Pro for Macintosh. His interests include developing games, software utilities, and application class libraries. He can be reached at jeremyv@farallon.com.

QC can be set up to run tests and suites of tests on a "target" or "target program". Target programs don't have to be applications; they can be other executable code such as control panels. Test suites include tests that validate blocks of memory, tests that detect common Memory Manager errors, and tests that stress the application heap to ensure that the application code runs properly under difficult memory conditions.

### How Does It Work?

Unless you are using the API, QC performs all of its tests by patching traps. QC will only report errors related to the tests that have been enabled for the target program, and it will not report errors when it is not running. QC reports every error with a `_DebugStr` call which will drop into MacsBug or your high-level debugger if it is set up to do so. In addition to reporting the error, QC reports the address of the last trap called in your code before the error occurred, allowing you to trace through the code to chase down the bug.

QC can automatically launch itself along with your target. You can choose to have QC launch when the target program calls `_InitGraf` or when a certain resource is loaded. Making QC wait until a resource is loaded is useful for non-applications such as control panels that will not call `_InitGraf`.

A key combination can be set that will launch and unlaunch QC while the target program is running. QC will pick the correct set of preferences for your target program when it is launched. As soon as QC is launched, the tests it performs are in progress.

Some tests in QC cannot distinguish when an error has occurred in an application other than the program being tested. It is therefore best to test your target program with only itself and the Finder running.

If you have all of the tests enabled, the slowdown caused by QC is quite noticeable, so be prepared. However, this is to be expected from such a low-level debugging tool, and is bearable.

### Memory Validation Suite

QC offers a suite of tests that validate the Memory Manager structures themselves, rather than the use of those structures. These tests are valuable for tracking down corrupted or incorrectly specified data.

The first test in the validation suite, **Cross-reference Master Pointers**, looks at all relocatable blocks in the heap (handles) and makes sure that they are pointed to by a master pointer within a non-relocatable block. This test can catch orphaning of relocatable blocks and corruption of the master pointer list.

The **Validate Handle/Pointer** test verifies that each handle/pointer is pointing to a block of memory with the correct properties (locked/unlocked for a handle, non-relocatable for a pointer) as they are passed to Memory Manager calls.


PLATFORMS: APPLE SYSTEM 7 • APPLE A/UX •
WINDOWS NT • WINDOWS 95 • OS/2 • NLM • DOS • LYNX • SCO • SYSTEM MANAGER • SUN SPARC-SOLARIS • IBM RS/6000
NT (ALPHA) • MIPS ABI (SGI) • AT&T UNIX • BANYAN • INTERACTIVE UNIX • LINUX • MOTOROLA 88OPEN • QNX •
• HP9000 • RS/6000 • SUN O/S 4.X •

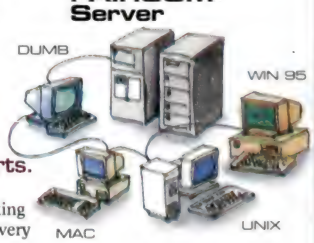
Q: What does it take to deploy a superior client/server application?
A: *A SUPERIOR SERVER*

**START** with the most advanced client-side SDK on the market: c-tree® Plus at $895.
- **Complete "C" Source code**
- ROYALTY FREE (Client Side)
- Multiple supported protocols
- Fast, portable, reliable
- Powerful features like transaction processing
- Win95, NT, and Windows 3.1 ready

**ADD** a strong, multi-platform, industrial-strength Server that supports.
- File mirroring
- Heterogeneous networking
- Automatic disaster recovery
- Multi-threaded design
- Best price/performance available: from $445- $3745

**RESULT?**
A solid, economical, easily deployable product that fits your needs.
- Portable
- Scalable
- Exceptional Performance
- Flexible
- Easy Server distribution
- Convenient OEM terms

FAIRCOM® Server




Heterogeneous TCP/IP Network

You can't find a better client SDK with these features!
Over sixteen years of proven reliability and performance.
No one else supports over 30 platforms in this price range!

c-tree Plus®
- Complete C Source
- Single/Multi User
- Client/Server (optional)
- Full ISAM functionality
- No Royalties
- Transaction Processing
- Fixed/Variable Length Records
- High Speed Data/Index Caching
- Batch Operations
- File Mirroring
- Multiple Contexts
- Unsurpassed Portability

FairCom® Server
- Client/Server Model
- Transaction Processing
- Requires <2MB RAM
- Online Backup
- Disaster Recovery
- Rollback - Forward
- Anti-Deadlock Resolution
- Client-side "C" Source
- Multi-threading
- Heterogeneous networking
- File Mirroring
- OEM/Source Available

FOR YOUR NEXT PROJECT CALL FAIRCOM: YOU CAN'T FIND A BETTER HETEROGENEOUS CLIENT/SERVER SOLUTION!

Also inquire about these FairCom products:
d-tree™
r-tree®
ODBC Driver

FAIRCOM® CORPORATION
*Since 1979*
WWWeb Address: http://www.faircom.com/
800-234-8180
**U.S.A.** phone (573) 445-6833 fax (573) 445-9698
**EUROPE** phone (035) 773-464 fax (035) 773-806
**JAPAN** phone (0592) 29-7504 fax (0592) 24-9723

# Powerful Email Publishing

# LISTSTAR®

Begin sponsoring your own email discussion groups and Email-On-Demand services with the award-winning ListSTAR for Macintosh. The services you can create are only limited by your imagination:

**Email Discussion Groups:**
- Maintain beta test lists
- Share tech support information
- Communicate with intra-office discussions
- Create software announcement groups
- Start a personal discussion group
- Publish discussions on the Web

**Email-On-Demand Services**
- Automatically reply to email inquiries
- Offer email technical support
- Send software updates to registered users
- Serve corporate information internally

**Download your FREE evaluation copy:**
http://www.starnine.com/software/

**Special Developers Price**
**Only $399.00**

(POP3 version also available for $99)
Offer good through September 1, 1996

2550 Ninth St, Suite 112
Berkeley, CA 94710
1-800-525-2580 or FAX (510) 548-0393

**1-800-525-2580**
**http://www.starnine.com/liststar/**


### Memory Manager Misuse

QC provides the programmer with a series of tests that detect common misuses and mistakes made while using the Memory Manager. It is the commonness of these problems that makes these tests so useful in debugging code.

Activating the **Detect Write to Zero** test checks, on each trap call, as to whether your target program has overwritten location zero. QC cannot determine whether it was your target program or another running application that overwrote zero, so again it is best to test with as few applications open as possible.

The **Dereferencing Zero** test puts data into location zero that will cause a bus error when a nil pointer/handle is dereferenced by the target program (a very common mistake).

**Reasonable Allocations** checks to make sure that allocations of data with `_NewHandle`, `_NewPtr`, etc., are passed reasonable values for the data size. Unreasonable allocations might include a negative data size or a data size that exceeds the size of the application heap.

QC's ability to detect when a resource is disposed of incorrectly with `_DisposeHandle` or a handle is disposed of incorrectly with `_ReleaseResource` is a very useful feature. This is probably one of the mistakes most easily made by Mac programmers when writing code that uses both handles and resources. Enabling this test almost immediately showed me places in my (untested) code where I wasn't releasing resources with `_ReleaseResource`, causing memory leaks and a possibly damaged resource map.

**BlockMove Checking** checks the values passed to `_BlockMove` to make sure that all addresses and data sizes are valid. With **Block Bounds Checking** enabled, QC adds a small tag value to every allocated block. QC periodically checks to see if this tag has been overwritten and reports an error if so. This is useful in catching errors related to writing past the boundaries of an array or a block of data.

**MemErr Detection** is a warning test that reports an error every time the low-level global `MemErr` is set to be anything other than `noErr`. This does not mean that the code is bad, but it can alert the programmer to places in the code where failed memory operations are not handled correctly. **Grow Lock** warns about attempts to resize a locked relocatable block. **Grow Non-Reloc** reports attempts to change the size of a pointer. Both of these operations are very likely to fail if a locked or non-relocatable block is ahead of the block being grown larger, but, as before, this may not represent an error if it is handled correctly in the code.

### Scrambled Heaps

Many times, a program is written, tested, and approved under ideal memory conditions, but when someone else uses the software with a lowered memory partition and several other applications running simultaneously, the program cannot handle the stress. It can be difficult to simulate these stressful conditions if you don't know what to look for. Of course, you can always lower the application memory size yourself and

launch several other applications every time you test your own program, but this is a difficult solution at best. QC offers a seamless method of ensuring that all of your code will continue to operate under even the most stressful memory conditions.

**Scramble Heap** continuously moves all relocatable blocks of memory. If you use a pointer to point to the data represented by a relocatable block and that block is moved in memory, your pointer will no longer be valid. Scrambling the heap is a good way to make sure your code handles this possibility.

**Purge Heap** will purge handles marked as "purgeable" whenever possible. This is a useful tool for detecting sections of your code where you did not think it possible for a handle to become purged, but it has. This is a likely situation if you have always tested your application with lots of extra memory, but the user lowers the memory partition.

**Check Heap** checks the structure of the heap before every Memory Manager call to ensure its reliability. **Invalidate Free Memory** fills all free blocks with a value that will cause a bus error if dereferenced by the target program. This will prevent you from accessing a block of data that has been released from memory.

## QC API

The QC API provides a set of routines in C for accessing directly from the target program code the individual tests that QC is able to perform. This is useful for making a special "debug" build of a product that routinely checks for memory-related errors near crucial portions of code.

Any of the tests that can be enabled from the QC control panel can also be enabled with the QC API. QC must be installed for the QC API to work.

## Conclusion

QC is ideal for small developers who don't have the resources to write tools to extensively stress-test their products. Larger developers are more likely to have their own customized stress-testing tools, but QC may fill the gap for those memory-testing features that are missing.

If you are a small developer who doesn't have the time to track down memory-related bugs after your product ships, QC is worth the money. Buy it. If your program has problems, QC will alert you to them and will protect you from being embarassed by unforeseen glitches in your code. If your program is spotless, QC will provide you with that needed bit of extra security (and extra sleep) when your product is shipped. For only $100, it is a worthwile investment in the quality of your products and the satisfaction of your customers.

*MacTech's resident programmer, Don Bresee, adds: "QC gives you the ability to consistently reproduce the nastiest of bugs, which is always the first step to finding bad code. QC immediately helped me find a bug in THINK Reference that was no less than four years old."*

**The Ultimate Mac Web Server**

**WEBSTAR®**

Join the growing number of people serving Web sites on their Macintosh. WebSTAR, voted "Software of the Year" from *MacUser* Magazine, lets you easily create a powerful and dynamic Web site for the Internet or Intranet:

- **New!** Plug-in architecture for enhanced performance and extensibility.
- Remote adminstration over TCP/IP network.
- Better cost/performance and more secure than many UNIX machines.
- FREE Adobe PageMill, FTP, Gopher servers

**Download your FREE evaluation copy:**
http://www.starnine.com/software/

**Special Developers Price only $399.00**
Offer good through September 1, 1996

**Developers:**
The WebSTAR API for creating your own plug-ins is now available. This new architecture makes writing robust add-on extensions extremely easy. To receive your copy of the WebSTAR API, please point your browser to:
http://www.starnine.com/webstar/webstarsdk.html

2550 Ninth St. Suite 112
Berkeley, CA 94710
1-800-525-2580 or FAX (510) 548-0393

**1-800-525-2580**
**http://www.starnine.com/webstar/**

# PROGRAMMER'S CHALLENGE

*By Bob Boonstra*

## A-MAZE-ING

Deep underground in an abandoned mine. No lantern. No rope. Not much air. Or something like a lack of air that makes you want to leave in a hurry. This month you find yourself lost in an underground maze with nothing except your trusty Challenge solution to help you quickly find your way out.

The prototype for the code you should write is:

```
typedef Boolean    /* found exit */ (*MoveProc) (
   long xMove,     /* amount of attempted move in x direction (1, 0, or -1) */
   long yMove,     /* amount of attempted move in y direction (1, 0, or -1) */
   long zMove,     /* amount of attempted move in z direction (1, 0, or -1) */
   long *newXPos,  /* new x position after attempted move */
   long *newYPos,  /* new y position after attempted move */
   long *newZPos   /* new z position after attempted move */
);

Boolean /* found exit */ Maze (
   long xPos,              /* x component of initial position */
   long yPos,              /* y component of initial position */
   long zPos,              /* z component of initial position */
   long xSize,             /* size of maze in x dimension */
   long ySize,             /* size of maze in y dimension */
   long zSize,             /* size of maze in z dimension */
   MoveProc MakeAMove,     /* callback to attempt a move */
   char *mapStorage        /* one byte for each position in the maze */
);
```

Your `Maze` routine will be called with your initial position (`xPos,yPos,zPos`) and the address of a callback routine you can use to attempt to move within the maze. Your code should make sequential calls to the `MakeAMove` callback until the callback returns `TRUE`, indicating that you have found your way out of the maze. All exits are on the boundaries of the maze (i.e., at least one coordinate equals `0` or `[xyz]Size-1`). With each move, you can attempt to move up to one unit in each dimension (`x`, `y`, and/or `z`). The callback will return your new position (`*newXPos,*newYPos,*newZPos`) after the attempted move. If a passageway exists in the desired direction, the new position will be offset from your previous position by the desired amount. If no passageway exists, the new position will be the same as the previous position (and you will be slightly bruised from walking into solid rock). One slight complicating factor is the combination of darkness and gravity – if you walk into a position where a vertical shaft is open below you, the callback will return a position where the vertical (`z`) coordinate is reduced the distance you fell. For this reason, moves in a z-only direction are never needed, as a purely downward move will always fail (you will already have

### THE NEW DEADLINE

The Challenge is due on the **first** of the month, not the tenth, as has been the case since the column started. To get the Challenge problem in time, you should be subscribed to the Programmer's Challenge Announcement List.

Subscribe by sending an email to "macjordomo@listmail.xplain.com", with the **message** "sub challenge-a Your Name".

There's also an open discussion list to discuss Challenges, review solutions, and talk about efficient computing. You subscribe to this list by sending an email to "macjordomo@listmail.xplain.com", with the **message** "sub challenge-d Your Name".

And finally, there's an ftp site for members of the Challenge lists. Its URL is:
ftp://ftp.mactech.com/mactech/CHALLENGE.

People have used the ftp site to trade sample data and upload alternative solutions.

### THE RULES

Here's how it works: Each month we present a new programming challenge. First, write some code that solves the challenge. Second, optimize your code (a lot). Then, submit your solution to *MacTech Magazine*. We choose a winner based on code correctness, speed, size, and elegance (in that order of importance) as well as the submission date. In the event of multiple equally desirable solutions, we'll choose one winner (with honorable mention, but no prize, given to the runner up). The prize for each month's best solution is a $100 credit for Developer Depot™ and a limited-edition "The Winner! MacTech Programmers Challenge" T-shirt (not available in stores anywhere).

Unless stated otherwise in the problem statement, the following rules apply: All solutions must be in ANSI compatible C. Use only pure C code. We disqualify entries with any assembly in them (except for challenges specifically stating otherwise). You may call any Macintosh Toolbox routine (e.g., it doesn't matter if you use NewPtr instead of malloc). We test entries with compiler options set to disable FPU use (for 680x0 code) and to enable all available speed optimizations. The compiler to be used and the target instruction set (680x0 or PowerPC) will be stated in the problem. **Limit your code to 60 characters per line;** this helps with e-mail gateways and page layout.

We publish the solution and winners for each month's Programmer's Challenge two months later. All submissions must be ***received by*** the 1st day of the month printed on the front cover of this issue.

You can get a head start on the Challenge by reading the Programmer's Challenge mailing list. It will be posted to the list on or before the 12th of the preceding month. To join, send an email to macjordomo@listmail.xplain.com with the message "sub challenge-A YourName'.

Mark solutions "Attn: Programmer's Challenge Solution" and send it by e-mail to one of the Programmer's Challenge addresses in the "How to Communicate With Us" section on page 2 of this issue. Include the solution, all related files, and your contact info.

*MacTech Magazine* reserves the right to publish any solution entered in the Programmer's Challenge. Authors grant *MacTech Magazine* the exclusive right to publish entries without limitation upon submission of each entry. Authors retain copyrights for the code.

# Developer Central™

Sponsored by

and

*For Macintosh Programmers & Developers*

MacTech™ MAGAZINE

**Macworld Expo/Boston** • August 7 – 10, 1996
Bayside Expo Center

## Come to the Center of the Universe!

*(The Macintosh development universe.)*

*If you are developing custom applications,* custom solutions, multimedia, or engineering solutions for the Macintosh, then you need to visit Developer Central. Here – all in one place – you will see the latest tools for Macintosh and Newton development. In addition, Apple Computer will be available with information on its training and programs to support developers.

*If you want to jump-start your knowledge* of Macintosh development or come up to speed on a new technology or tool, then visit the Developer Central Theater, where representatives of Apple's Developer University and others will be presenting the inside scoop on Macintosh development.

## At Developer Central, you can…

- Talk to representatives and demo products from a number of Third Party Vendors
- Attend sessions in the Developer Central Theater
- Get the best deals on tools from Apple Developer Catalog and Developer Depot™

So, no matter if you are corporate developer putting together a departmental system,
use the Macintosh in science,
or are aiming to have the next great Macintosh application or multimedia title,
*you will find what you need at Developer Central.*

## Developer Central
## The Source for Macintosh Development

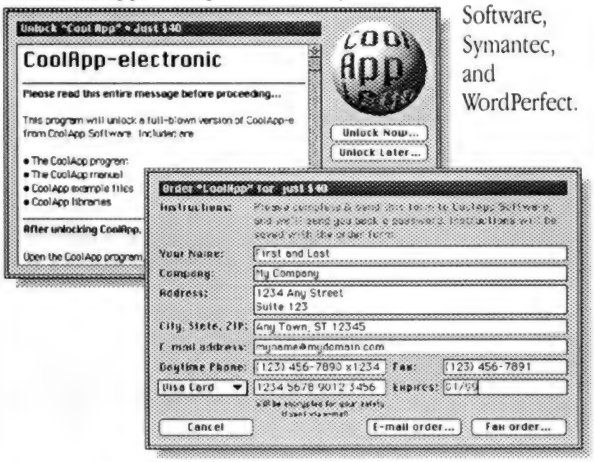
**Sponsored by Apple Computer, Inc. and MacTech Magazine**

Apple and the Apple logo are trademarks of Apple Computer, Inc. MacTech, MacTech Magazine and Developer Depot are trademarks of Xplain Corporation.
Developer Central and Virtual Developer Central are trademarks of Xplain Corporation.

**SOLID SOLUTION FOR ELECTRONIC DISTRIBUTION**

# BroadCast™

BroadCast is a highly secure, patent-pending system for safely distributing "locked" software online, or on CD-ROM. This cutting-edge technology was developed by the same engineering team behind PatchWorks— our tool for creating online updaters that's become an industry standard used by clients like Apple Computer, Adobe Systems, MacroMind, Now Software, Symantec, and WordPerfect.




**BROADCAST IS EASY FOR YOU AND YOUR CUSTOMERS**

In seconds, BroadCast securely locks your product into an Unlocker application that can be safely distributed to the public. To buy your product, customers simply run the Unlocker— then either call you, or complete an onscreen order form that's emailed or faxed to you with a unique "control number" (and encrypted credit card data).

Your staff processes the credit card, registers your new user, and provides your customer with a password based upon the control number. To unlock your software, the customer enters this password in the Unlocker.

**BROADCAST IS INEXPENSIVE**

We aren't resellers or distributors. Customers buy directly from you, and we take no percentage of your sales— just a small fee per transaction. In fact, BroadCast is distributed for free. We're betting you'll quickly use the free password points that come with BroadCast, and purchase more as your sales soar.

We'll even help attract customers by advertising your products for free on our "Software Unboxed" internet mall.

**GET THE WHOLE STORY**

We invite you to learn more by visiting our internet web site at "http://www.broadcastsoft.com", and download your free BroadCast Starter Kit, or email us: "info@broadcastsoft.com".

BroadCast

**BroadCast Software Corp**
East: 407-241-0308, Fax: 407-241-3195
West: 503-317-0429, Fax: 503-317-0430
Net: http://www.broadcastsoft.com
Email: info@broadcastsoft.com


fallen), and a purely upward move would be futile (you would fall back to your previous position).

The callback will obviously know which positions in the maze are filled with rock and which are open. In the event you wish to construct a map as you grope around in darkness, the `mapStorage` parameter will point to `xSize*ySize*zSize` bytes of storage for your use, initialized to zero by the caller. Once the callback indicates that you have found an exit for the maze, you should return `TRUE`. If you cannot find an exit, you should return `FALSE` (rather than loop forever), although this should not happen, because a path to an exit will always exist.

Selection of the winning solution will be based on total execution time of the `Maze` routine, including the time spent in the callback. The same callback code will be used to test each solution, providing a fixed time penalty for each attempted move.

This will be a native PowerPC Challenge, using the latest CodeWarrior environment. Solutions may be coded in C, C++, or Pascal. To keep the contest focused on code efficiency (rather than compiler efficiency), all solutions will be scored using the Metrowerks compiler for the appropriate language.

### Next Month – PowerPC Assembly Language

Next month is the annual assembly language Challenge, and this will be the first year that the Challenge will be in PowerPC assembly language. Depending on the difficulty of the problem, I may provide the electronic version of the problem statement a little early to allow some extra time for you to code your solutions. (See "The New Deadline", on the previous page, for instructions on subscribing to the Challenge mailings.)

### Two Months Ago Winner

Congratulations once again to **Ernst Munter** (Kanata, ON), for submitting the fastest entry to the Programmer's Challenge. This Challenge called for implementation of a comparison-free distribution sort algorithm that, in theory, can execute in time that grows linearly with the number of input records. While Ernst won the Postman's Sort Challenge by default, as the only person to submit an entry, his solution is both efficient and instructive.

The problem statement required that the solution deal with cases when memory was scarce, requiring the distribution sort being performed using temporary disk files. Ernst's solution also took advantage of the case when ample storage was available. The table below summarizes the results for a test case of 100000 records, given available working storage of 64KB, 640KB, 1MB, and 10MB, respectively. Numbers in parentheses after Ernst's name indicate his cumulative point total for all previous Challenges, not including this one.

| | | | | time (seconds) | | | |
|---|---|---|---|---|---|---|---|
| **Name** | **size** | **data** | **Memory:** | **64K** | **640K** | **1MB** | **10MB** |
| Ernst Munter (174) | 4312 | 1112 | | 2314 | 221 | 201 | 18 |

I examined a few other cases to indicate the growth in sort time as a function of the number of records. Much more

information would be needed for a definitive conclusion, but the growth appears to be somewhere between the $O(N\log_2 N)$ behavior of comparison sort algorithms and the theoretical best case linear behavior of a distribution sort. I've included the results for a case where the temporary sort files were written to a RAM disk, which clearly indicates the dominant role of disk I/O time.

| | time (seconds) | | |
|---|---|---|---|
| **Scenario / Memory available** | **640K** | **1MB** | **10MB** |
| 20000 records | 64 | 36 | 3 |
| 100000 records | 221 | 201 | 18 |
| 500000 records | 1912 | 1514 | 644 |
| 20000 records (RAM disk) | 9 | 5 | 2 |

With this win, Ernst Munter takes over **first place** in the Programmer's Challenge points standing. Congratulations, Ernst!

Ernst has vaulted into first place by winning four of the last five Challenge contests. To give greater recognition to contestants like Ernst who are currently participating in the Challenge, I've decided to revise slightly the Top Contestants table that we publish each month. From now on, the published point total will include points earned in the twenty-four most recent contests. With this new scheme, a single first-place finish is currently enough to earn you a spot in the Top Contestants table.

### Top Twenty Contestants

Here are the top twenty contestants for the Programmer's Challenge. The numbers below include points awarded over the twenty-four most recent contests, including points earned by this month's entrants.

| Rank | Name | Points | Rank | Name | Points |
|---|---|---|---|---|---|
| 1. | Munter, Ernst | 187 | 11. | Vineyard, Jeremy | 22 |
| 2. | Gregg, Xan | 92 | 12. | Cutts, Kevin | 21 |
| 3. | Larsson, Gustav | 87 | 13. | Picao, Miguel Cruz | 21 |
| 4. | [Name deleted] | 74 | 14. | Brown, Jorg | 20 |
| 5. | Lengyel, Eric | 40 | 15. | Gundrum, Eric | 20 |
| 6. | Lewis, Peter | 30 | 16. | Karsh, Bill | 19 |
| 7. | Darrah, Dave | 29 | 17. | Stenger, Allen | 19 |
| 8. | Beith, Gary | 24 | 18. | Mallett, Jeff | 17 |
| 9. | Kasparian, Raffi | 22 | 19. | Nevard, John | 17 |
| 10. | Noll, Robert | 22 | 20. | Nicolle, Ludovic | 14 |

There are three ways to earn points: (1) scoring in the top five of any Challenge, (2) being the first person to find a bug in a published winning solution, or (3) being the first person to suggest a Challenge that I use. The points you can win are:

1st place .....20 points
2nd place.....10 points
3rd place .....7 points
4th place.....4 points
5th place.....2 points
finding bug.....2 points
suggesting Challenge...2 points


**Good Things Come in SMALLER Packages**

Why have more than 700 developers selected Smaller Installer to deliver their products to customers around the world?

- ✓ Simple user interface
- ✓ Customizable windows and splash screens
- ✓ No scripting required
- ✓ Compact Pro® compression technology minimizes number of diskettes required
- ✓ Installer uses less than 40K of disk space
- ✓ Locates and installs files in system-related folders (Extensions, Fonts, etc.)
- ✓ Converts fat binaries to machine-specific versions during install
- ✓ Uninstall capability
- ✓ Password protection
- ✓ Extendable operation using your custom code resources
- ✓ Foreign language support – French, German, Japanese and Spanish

Smaller Installer is licensed with a one-time fee. No annual fees. Prices vary from $175 to $1,000 depending on quantity shipped.

Call or visit our web site to obtain a free developer kit which includes everything necessary for creating your own installers.

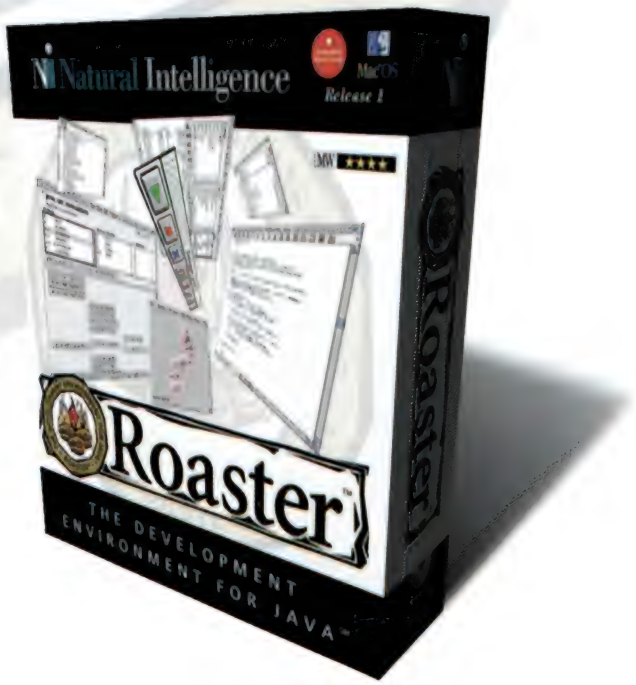
Smaller Installer™

**Cyclos**
P.O. Box 31417
San Francisco, CA 94131
415/821-1448
415/821-1168 (fax)
sales@cyclos.com
http://www.cyclos.com

JAVA

MACINTOSH

ROASTER




MW ★★★★

*Macworld - May '96*

**orders: (800) 739-1616**
**email: orders@natural.com**
**web: www.roaster.com**
**international: (617) 556-2588**

Natural Intelligence

www.natural.com, info@natural.com

Roaster is a trademark of Natural Intelligence Inc. All rights reserved. Copyright © 1995-1996 Natural Intelligence Inc. and its licensors. Macintosh is a registered trademark of Apple Computer Inc. Mac is a trademark of Apple Computer Inc. Java and Sun are trademarks of Sun Microsystems Inc. All trademarks referenced are the trademarks or registered trademarks of their respective holders.

Here is Ernst's winning solution:

**POSTPLUS.CPP**

Copyright © 1996 Ernst Munter, Kanata, ON, Canada

```
/*
    "Postman's Sort"

Problem Statement
-----------------

Sort an address file, using distribution sort, and avoid any direct or disguised
(indirect) comparison between sort keys of the input records.

Solution
--------

It was tempting to create a sorted index tree of the keys, and then sort input records
by comparing (descending) the sort tree. However, I think that would not have been
in the spirit of the Challenge, which was, I believe, to find a fast implementation of
Bob Ramey's postman's sort algorithm. Hence, this solution works entirely by
distributing records into bins, or stacks, directly indexed by a single character.

Tabs, and end-of-record markers are converted to \x0, the range \x1 to \x1F is used
to sort numeric fields by their length. Characters above 'a' are reduced by 0x20,
which makes all upper and lower case characters the same. Coincidentally, it also
sets {|}~ equal to [\]^.

Whether the input file fits in memory or not, the principle of sorting is the same:

All records are parsed into nodes and attached to stacks based on the current sort
key, for example the first letter of country. Then each stack is recursively sorted on
the next sort key, i.e. the second letter of country, in the same manner.

If this is done in-memory (MemSort) the stacks are linked and detached from the
global STACK array, before sorting each stack.

If the stack was too large to fit in memory, it is sent in segments to a temporary file
on disk. The temp file then contains interleaved segments of stacks which we keep
track of in a local variable (bin).

To continue sorting, the segments of a particular stack are retrieved from disk, and
the whole stack may now fit in memory for final sorting. If it does not, it is again
sorted into segments and sent to a new temp file on disk.

The number of temp files needed is equal to the recursion depth of the disk sort, until
a stack can be sorted to the finish in memory. The more memory we have, the fewer
temp files are needed.

Assumptions
-----------

tmpfile() is used to create temporary files. Depending on system setup, the system
may run out of files to open, and sorting will then stop.

The order of numerical fields will be natural numerical, that is 9, 12, 30, 100.
However, leading zero's are not expected (or we get 41, 80, 053, 102).

The length of a purely numeric field to be handled correctly should not exceed 31.
Longer numeric fields could get sorted behind fields starting with punctuation.

The longest record expected should be substantially less than the constant MINMEM
which is set to 1024, but this limit can be relaxed. It plays a role only when we are
very short of memory.

All ASCII characters may occur, and will be sorted on. Any control characters other
than Tab (0x09) will be treated as end-of-record indicators.

The sort is not "stable", in the sense that the order of equivalent records (differing in
case only, as an example) is not preserved. Records are tossed on stacks and into
bins, and are frequently retrieved in reverse order. Furthermore, running the program
with different amounts of storage may result in a different order as well (U/L case).
*/

#include <stdlib.h>
#include <stdio.h>
#include <memory.h>
#include <string.h>

#define uchar unsigned char
#define ushort unsigned short
#define ulong unsigned long
```

```
#define NUM_KEYS  224
#define MINMEM    1024
#define MIN_BUFFER  512
#define DEF_BUFFER_SIZE 16384
#define numFields 9
#define lastField (numFields-1)
#define TAB_CHAR  0x09
#define END_RECORD  0x0D

typedef struct Node Node;
struct Node {
  uchar* ref;
  Node*  next;
  Node*  link;
  ushort fld[numFields];
  ushort len;
};

typedef struct tHeader tHeader;
struct tHeader {
  long next;
  int  size;
  int  key;
};

typedef struct Bin Bin;
struct Bin {
  FILE* tf;         //temp file
  long next;        //next segment in temp file
  int  bytesLeft;  //in current segment
  long fpos[NUM_KEYS];
};

typedef struct OutputBuffer OutputBuffer;
struct OutputBuffer {
  int  space;       //space left
  int  size;        //size of buffer
  uchar c[4];       //will be c[size]
};

static FILE*    theOutFile;
static OutputBuffer* buffer;
static int      topKey;
static Node*    STACK[NUM_KEYS];
```

Macros and prototypes

```
// Output buffer functions:
// The output buffer accumulates characters to be sent to the output file. The block
// size is chosen as 16K, but reduced to 8K if less than 128K of storage is available.
// The buffer isreleased if, after many recursions storage space is severely reduced and
// we need the memory for sorting.

void FlushBuffer();
int  OpenBuffer(uchar* storage,int storageSize);
void CloseBuffer();
void BufferedWrite(uchar* text,int size);
void BufferedSend(Node* node);

// in-memory sort and parsing functions:

Node*  Link(int* numeric);
int  TestNumeric(int key,uchar* text,int len);
void Dist(Node* node,int field,int offset,int test);
void MemSort(int field,int offset);
int  Parse(uchar* text,Node* node,
           int size,int field,int offset,int test);

// SStream functions:
// SStreams are virtual files which are interleaved on a single disk file.
// Each SStream may consist of several segments, linked together through small
// headers (tHeader) on the disk. The last segment is pointed to from a "bin".
// Bins administer access to these streams.

void InitBin(Bin* bin);
int  OpenBin(Bin* bin);
void ResetBin(Bin* bin);
void CloseBin(Bin* bin);
int  SendToTemp(Bin* bin);
int  SegmentSize(Node* node);
void Send(Node* node,FILE* f);
int  SStreamSeek(Bin* bin,int key,long pos);
int  SStreamRead(uchar* text0,int size,Bin* bin);
long SStreamSize(Bin* bin,int key);
void SendSStreamOut(uchar* text,int storageSize,Bin* bin,int
                                    key);

// The DiskSort function is called from PostmansSort, and calls itself recursively.
// With each recursion, the calling Bin structure must be preserved which reduces the
// storage available for the called DiskSort function. This wastage is somewhat
// controlled by recovering the unused part of the Bin structure, since keys in the high
// ASCII ranges would frequently not be needed.

// Bins could also have been put on the procedure stack, but this could put a more
// severe limit on recursion depth. (about 1K per Bin), depending on the environment.

int  RecoverMemory();
int  DiskSort(Bin* inBin, uchar* storage, int storageSize,
              int field, int offset);

#define QUIT(rc) {CloseBin(bin); return rc;}
#define EMPTY_BIN(bin,key) (bin->fpos[key]<0)

/* Ernst included shortCut functions used to detect the case of a single key in a stack
 of records. These functions were not used in the evaluation, and have been deleted
 for length. They can be found in the online version of the solution. [JRB] */
```

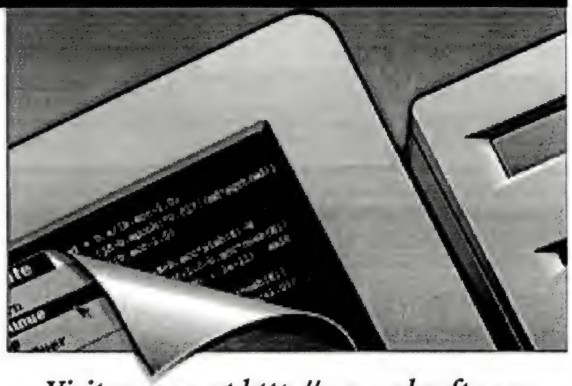
PostmansSort

```
pascal OSErr PostmansSort(
  FILE *inFile,          /* stream containing sort input */
  FILE *outFile,         /* stream to contain sort output */
  void *privateStorage,  /* preallocated storage for your use */
  size_t storageSize     /* number of bytes of privateStorage */
) {
uchar* storage=(uchar*)privateStorage;
Bin* bin=(Bin*)storage;
uchar* text;
Node*  node;

long fileSize,shortage,curPos;
int  textSize,recovered,rc;
int  nextField=lastField;

// Setup storage:

  text=(uchar*)(bin+1);
  theOutFile=outFile;
  storageSize-=OpenBuffer(storage,storageSize);
  storageSize-=sizeof(Bin);
  storageSize-=sizeof(Node);
  if (storageSize<MINMEM) return 10;

// Determine size of input file and read as much as possible

  curPos=ftell(inFile);
  fseek(inFile,0,2);
  fileSize=ftell(inFile);
  fseek(inFile,curPos,0);
  textSize=fread(text,1,storageSize,inFile);

// Parse input data. Nodes grow from the top of storage
// down into the text, and may cause some of it to be reloaded

  node=(Node*)(text+storageSize);
  topKey=0;
  shortage=Parse(text,node,textSize,nextField,0,1);
  curPos=textSize-shortage;

  if (fileSize-curPos==0) {  //everything fit

    MemSort(nextField,0);

  } else {       //distribute into bins

    InitBin(bin);
    do {
      if (0!=(rc=SendToTemp(bin))) QUIT(rc);
      fseek(inFile,curPos,0);
      textSize=fread(text,1,storageSize,inFile);
      node=(Node*)(text+storageSize)-1;
      shortage=Parse(text,node,textSize,nextField,0,1);
      curPos+=textSize-shortage;
```

# World's Leading FORTRAN 77 for Macintosh

## 68K and Power Macintosh versions

- *full ANSI 77 native compiler*
- *faster execution speed*
- *graphical source-level debugging*
- *two complete graphics packages*
- *make utility, MRWE application framework*
- *System 7.5 compatible, MPW included*
- *Windows 95/NT version also available*



absoft
development tools and languages

*Visit us now at http://www.absoft.com*

2781 Bond Street Rochester Hills MI 48309 • (810) 853-0050 • Fax (810) 853-0108 • sales@absoft.com


```
    } while(curPos<fileSize);
    if (0!=(SendToTemp(bin))) QUIT(rc);
      {
        recovered=RecoverMemory();
        if (0!=(rc=DiskSort(bin,text-recovered,
              storageSize+recovered,nextField,0)))
                QUIT(rc);
      }

    CloseBin(bin);
  }

  FlushBuffer();
  return 0;
}
```

InitBin

```
void InitBin(Bin* bin) {
  memset(bin->fpos,-1,sizeof(bin->fpos));
  bin->next=0;
  bin->bytesLeft=0;
  bin->tf=0;
}
```

OpenBin

```
int OpenBin(Bin* bin) {
  if (bin->tf==0) {
    if (0==(bin->tf=tmpfile()))
      return 5;   // unable to open temp file
  }
  return 0;
}
```

ResetBin

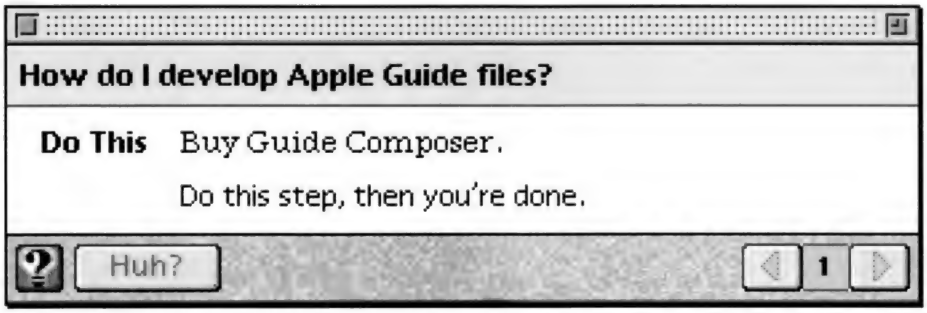
```
void ResetBin(Bin* bin) {
  memset(bin->fpos,-1,sizeof(bin->fpos));
  bin->next=0;
  bin->bytesLeft=0;
  if (bin->tf) fseek(bin->tf,0,0);
}
```

CloseBin

```
void CloseBin(Bin* bin) {
  if (bin->tf) {
    fclose(bin->tf);
  }
}
```

Parse

```
int Parse(uchar* text, Node* node, int size, int field,
                int offset,int test) {

// returns the shortage, i.e. the difference between the end of the loaded text and the
// end of the last record parsed.  The shortage is caused by the fact that we load text
// up to the storage limit, but then overwrite some of it by the parsed nodes which
// grow downwards from the top. In this way we don't have to guess at the size of a
// typical record.

uchar* record=text;
int  xfield=0;
uchar* textEnd=text+size;
  while (text<(uchar*)node) {
    int c=*text++;
    if (c<' ') {
      if (c==TAB_CHAR) {

// mark off those fields
          xfield++;
          if (xfield>lastField) return 4;
          node->fld[xfield]=(ushort)(text-record);
        } else {

// any other CTL-char is treated as END_RECORD:
          int key=*(record+node->fld[field]+offset);

// fill in fields of node and hook to STACK:
          node->ref=record;
          node->link=0;
          node->fld[0]=0;
          node->len=(ushort)(text-record);

// key is the character in the current key position modified to
//  0 if end of field,
//  lower case becomes upper case
//  length of field if numeric field detected

          if (node->len==1) key=0;
          else if (key>='a') key-=0x20;
          else if (key<' ') key=0;
          else if (test)
            key=TestNumeric(key,node->ref+node->fld[field],
              node->fld[field+1]-node->fld[field]-1);

// keep track of the highest key value used, for future economies.

          node->next=STACK[key];
          if (key>topKey) topKey=key;
```




Available from the MacTech Mail Order Store

Demo/info:http:rampages.onramp.net/~stepup

$99

STEPUP SOFTWARE

7110 Glendora Avenue
Dallas, Texas 75230
214-360-9301
214-360-0127 fax

Guide Composer is a trademark of StepUp Software. Apple Guide is a trademark of Apple Computer, Inc.

```
            STACK[key]=node--;
            record=text;
            xfield=0;
        }
      }
      if (text>=textEnd)
        return 0;       //no shortage
    }

    return textEnd-record;
}
```

MemSort

```
static mms;

void MemSort(int field,int offset) {
int endField=(int)STACK[0],numeric=0;
Node* node;

  mms++;
  node=Link(&numeric);
  while (node) {
    Node* nextNode=node->link;
    if (0==node->next) {//no need to sort a single record
      BufferedSend(node);
      endField=0;
    } else if (endField) { //go to next field
      int nextField=field-1;
      endField=0;
      if ((field==7)&&(offset>0)) nextField-=2;
      Dist(node,nextField,0,1);
      MemSort(nextField,0);
    } else if (numeric) {  //test same key position for value
      numeric--;
      Dist(node,field,offset,0);
      MemSort(field,offset);
    } else {        //test next key position
      Dist(node,field,offset+1,0);
      MemSort(field,offset+1);
    }
    node=nextNode;
  }
}
```

Link

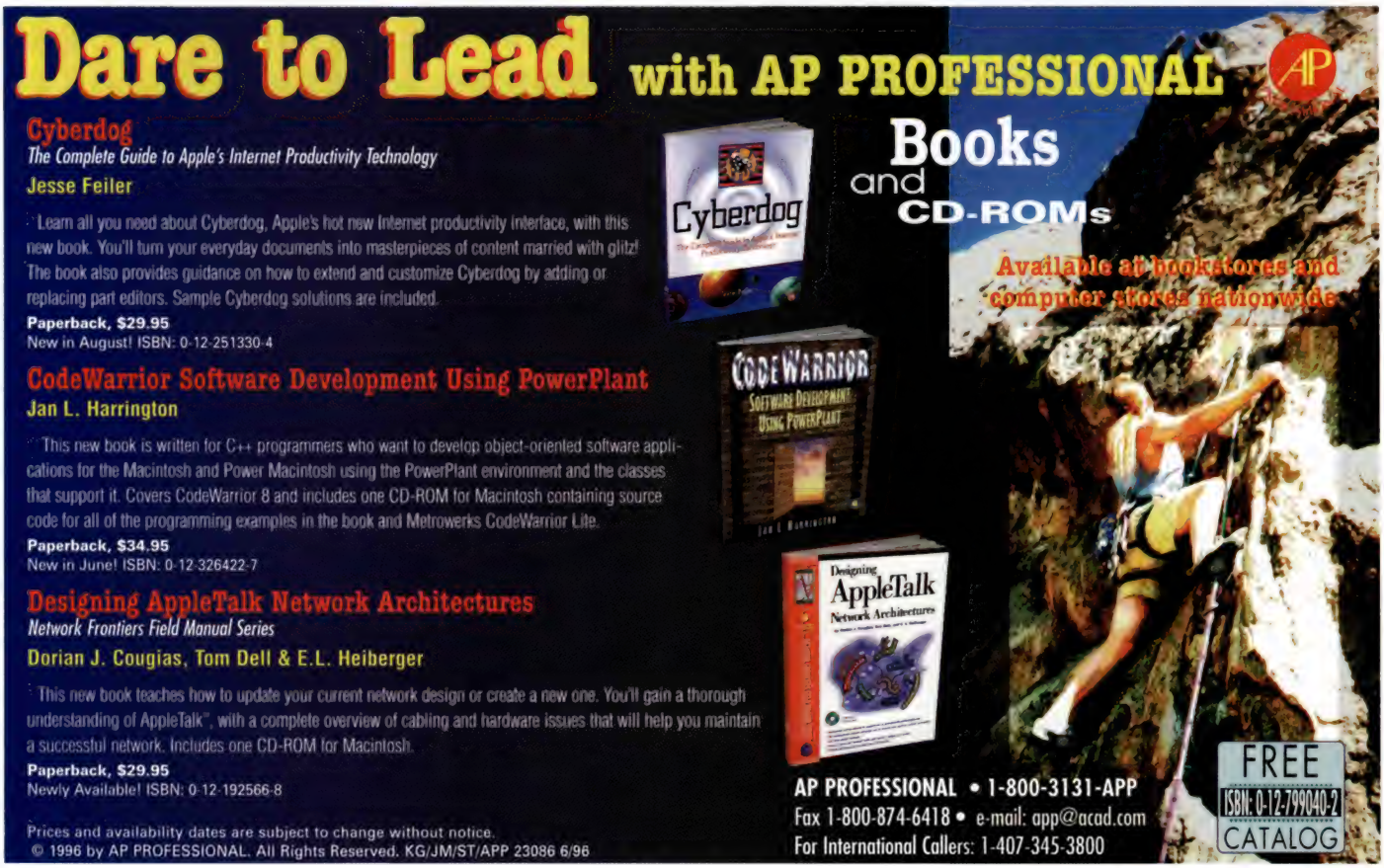
```
Node* Link(int* numeric) {
// detaches stacks under the same key from STACK[] and
// links their first nodes together
Node* node=0;
Node* n;
int key;
  for (key=topKey;key>=0;key--) {
    if (0 != (n=STACK[key])) {
      if ((key<' ') && (key>0)) (*numeric)++;
      n->link=node;
      node=n;
      STACK[key]=0;
    }
  }
  topKey=0;
  return node;
}
```

Dist

```
void Dist(Node* node,int field,int offset,int test) {
// scan all nodes in a stack and redistribute them
// back on STACK[] depending on the next key value.
  do {
    Node*  nextNode;
    int    key=*(node->ref+node->fld[field]+offset);
    if (key>='a') key-=0x20;
    else if (key<' ') {
      key=0;
      if (field==0) { //sorting done for this node
        nextNode=node->next;
        node->next=0;
  BufferedSend(node);
  node=nextNode;
  continue;
      }
    } else if (test) {
      key=TestNumeric(key,node->ref+node->fld[field],
        node->fld[field+1]-node->fld[field]-1);
    }
    nextNode=node->next;
    if (key>topKey) topKey=key;
    node->next=STACK[key];
    STACK[key]=node;
    node=nextNode;
  } while (node);
}
```

TestNumeric

```
int TestNumeric(int key,uchar* text,int len) {
// returns the field length of a numeric field, otherwise returns key unchanged
int i;
  for (i=0;i<len;i++) {
    int c=*text;
    if (c>'9') return key;
    if (c<'0') return key;
```


Dare to Lead with AP PROFESSIONAL Books and CD-ROMs
Cyberdog
The Complete Guide to Apple's Internet Productivity Technology
Jesse Feiler
Learn all you need about Cyberdog, Apple's hot new Internet productivity interface, with this new book. You'll turn your everyday documents into masterpieces of content married with glitz! The book also provides guidance on how to extend and customize Cyberdog by adding or replacing part editors. Sample Cyberdog solutions are included.
Paperback, $29.95
New in August! ISBN: 0-12-251330-4
CodeWarrior Software Development Using PowerPlant
Jan L. Harrington
This new book is written for C++ programmers who want to develop object-oriented software applications for the Macintosh and Power Macintosh using the PowerPlant environment and the classes that support it. Covers CodeWarrior 8 and includes one CD-ROM for Macintosh containing source code for all of the programming examples in the book and Metrowerks CodeWarrior Lite.
Paperback, $34.95
New in June! ISBN: 0-12-326422-7
Designing AppleTalk Network Architectures
Network Frontiers Field Manual Series
Dorian J. Cougias, Tom Dell & E.L. Heiberger
This new book teaches how to update your current network design or create a new one. You'll gain a thorough understanding of AppleTalk®, with a complete overview of cabling and hardware issues that will help you maintain a successful network. Includes one CD-ROM for Macintosh.
Paperback, $29.95
Newly Available! ISBN: 0-12-192566-8
Available at bookstores and computer stores nationwide
AP PROFESSIONAL • 1-800-3131-APP
Fax 1-800-874-6418 • e-mail: app@acad.com
For International Callers: 1-407-345-3800
FREE
ISBN: 0-12-799040-2
CATALOG
Prices and availability dates are subject to change without notice.
© 1996 by AP PROFESSIONAL. All Rights Reserved. KG/JM/ST/APP 23086 6/96


```
    text++;
  }
  return len;
}
```

DiskSort

```
int DiskSort(
  Bin* inBin,
  uchar* storage,
  int storageSize,
  int field,
  int offset) {

// recursively callable version of PostmansSort()

Bin* bin=(Bin*)storage;
uchar* text;
Node*  node;

long fileSize;
int  textSize;
long shortage;
long curPos;

int  key;
int  rc;
int  recovered;
int  topKeyIn=topKey;

  InitBin(bin);
  storageSize-=sizeof(Bin);
  storageSize-=sizeof(Node);
  if (storageSize<MINMEM) {
    if (buffer) {
      storageSize+=buffer->size+sizeof(OutputBuffer)-4;
      CloseBuffer();
      if (storageSize<MINMEM) return 8;
    } else return 9;
  }
  text=(uchar*)(bin+1);
  for (key=0;key<=topKeyIn;key++) {
    int nextField,nextOffset,numericTest;
    if (EMPTY_BIN(inBin,key)) continue;
    if (0==(fileSize=SStreamSize(inBin,key))) continue;

    nextField=field;
    nextOffset=offset;

    topKey=0;
    if (key==0) {    //end of field reached

      if ((nextField==7)&&(nextOffset>0)) nextField=4;

      else if (nextField==0) {//copy whole stream
          SendSStreamOut(text,storageSize,bin,0);
          continue;

      } else nextField--;

      nextOffset=0;
      numericTest=1;
    } else if (key<' ') {  //start of a numeric field
      nextOffset=0;
      numericTest=0;
    } else {               //a typical field
      nextOffset++;
      numericTest=0;
    }

    SStreamSeek(inBin,key,0);
    textSize=SStreamRead(text,storageSize,inBin);
    node=(Node*)(text+storageSize);
    shortage=Parse(text,node,textSize,nextField,nextOffset,
          numericTest);
```

# http://www.mactech.com

Netscape: Welcome to MacTech™ Magazine!
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
Location: http://www.mactech.com/
MacTech™
MAGAZINE
Visit our article archives and see 1230+ pages from the history of MacTech.
Macworld? Check out Developer Central!
About MacTech
Interact With MacTech
MacTech CD-ROM
Article Archives
File Archives
Mail Order
URL's
Mailing Lists
Jobs & Services Available
Upcoming Issues
Sprocket
Programming FAQ's
Special Messages:
Note: While most of this site is complete, you may see some things still under construction. Please excuse "the mess". :)
The Programmer's Challenge is now available on this web. Each month, programmers compete to produce the fastest, tightest code.
The new MacTech CD is shipping at a new low price of $49!
Did you go to Macworld but forgot to get information from a developer tool vendor? Check out information on Developer Central sponsored by Apple and MacTech.
You've found your way onto the MacTech Web™ site. To date, you'll find some 1500+ pages of material here. The focus of this site is to support the Mac™ OS programmer and developer community. Here you will not only find support for MacTech Magazine, but you also get access to an number of useful places on the net for Mac developers as well as information on how to get the latest Macintosh development tools.
Let us know what you think of the site. It's a living, breathing beast and we're here to improve it, change it, nurse it and grow it. Our webmaster address is almost always available at the bottom of each page on this site.
-- The folks at MacTech Magazine
10/2/95 - 15:21 (PST)
http://www.mactech.com/cgi-bin/map/mainMTlogo.map?11,-9

## (Need we say more?)

*For Macintosh Programmers & Developers*

MacTech™ MAGAZINE

# EightyRez

The editor for 'aete' resources

- Create AppleScript dictionaries easily
- View resources as collapsible outlines
- Cut and paste whole suites
- Create resources or Rez source files
- Apple Guide help included
- $40.00 plus $2.00 shipping
  (individual, corporate, & educational customers in USA)

CONCEPTUAL DESIGN

8 Ardon Drive
Hooksett, NH 03106
(603)485-5699

http://www.mv.com/biz/condes/EightyRez.html
E-mail: gmcgath@condes.mv.com


```
    curPos=textSize-shortage;

    if (fileSize-curPos==0) {   //sort in memory

      MemSort(nextField,nextOffset);

    } else {        //sort in temp files

      do {
           if (0!=(rc=SendToTemp(bin))) QUIT(rc);
           SStreamSeek(inBin,key,curPos);
           textSize=SStreamRead(text,storageSize,inBin);
           node=(Node*)(text+storageSize)-1;
           shortage=Parse(
                text,node,textSize,
                nextField,nextOffset, numericTest);
           curPos+=textSize-shortage;

      } while(curPos<fileSize);
      if (0!=(rc=SendToTemp(bin))) QUIT(rc);
      {
        recovered=RecoverMemory();
        if (0!=(rc=DiskSort(
                   bin, text-recovered,
                   storageSize+recovered,
                   nextField,nextOffset)))
                       QUIT(rc);
      }

      ResetBin(bin);
    }
  }
  CloseBin(bin);
  return 0;
}
```

SendToTemp

```
static sst;

int SendToTemp(Bin* bin) {
// send each segment (stack of equal key records) to disk,
// bin->fpos[key] points to the start of the last segment
// stored. The segment header points to the previous
// segment or -1 if this is the first.

int key,rc;
Node* node;
  if (0!=(rc=OpenBin(bin))) return rc;
  for (key=0;key<NUM_KEYS;key++) {
    if (0 != (node=STACK[key])) {
      tHeader th;
      long pos;
      STACK[key]=0;
      th.size=SegmentSize(node);
      th.next=bin->fpos[key];
      th.key=key+('#'<<8);
      pos=ftell(bin->tf);
      fwrite(&th,1,sizeof(tHeader),bin->tf);
      sst++;
      bin->fpos[key]=pos;
      Send(node,bin->tf);
    }
  }
  return 0;
}
```

SegmentSize

```
int SegmentSize(Node* node) {
// The length of a segment needs to be known for the header, before we store the text
// records. This functions scans through all nodes of a stack and adds up their lengths.
int s=0;
  do {
    s+=node->len;
    node=node->next;
  } while(node);
  return s;
}
```

Send

```
void Send(Node* node,FILE* f) {
// Sends a stack to disk, either the temp file, or
// the output file in the case of unbuffered output.
  do {
    fwrite(node->ref,1,node->len,f);
    node=node->next;
  } while(node);
}
```

SStreamSeek

```
int SStreamSeek(Bin* bin,int key,long pos) {
// returns error=7 if no SStream, or if seeked beyond size
long fpos=bin->fpos[key];
tHeader th;
  if (0==bin->tf) return 6;
  while (fpos>=0) {
    fseek(bin->tf,fpos,0);
    fread(&th,1,sizeof(th),bin->tf);
    if (pos<th.size) {
      fseek(bin->tf,pos,1);
      bin->bytesLeft=th.size-pos;
      bin->next=th.next;
      return 0;
    } else {
      pos-=th.size;
      fpos=th.next;
    }
  }
  return 7;
}
```

SStreamRead

```
int SStreamRead(uchar* text0,int size,Bin* bin) {
// reads from current position reached with SStreamSeek()
// returns the actual number of bytes read
long fpos;
tHeader th;
int bytes2read;
uchar* text=text0;
  if (bin->bytesLeft) {
    if (bin->bytesLeft>size) bytes2read=size;
    else bytes2read=bin->bytesLeft;
    fread(text,1,bytes2read,bin->tf);
    text+=bytes2read;    //even if fread comes short
    size-=bytes2read;
  }
```

```
  fpos=bin->next;
  while ((size) && (fpos>=0)) {
    fseek(bin->tf,fpos,0);
    fread(&th,1,sizeof(th),bin->tf);
    if (th.size>size) bytes2read=size;
    else bytes2read=th.size;
    fread(text,1,bytes2read,bin->tf);
    text+=bytes2read; //even if fread comes short
    size-=bytes2read;
    fpos=th.next;
  }
  return text-text0;
}
```

SStreamSize

```
long SStreamSize(Bin* bin,int key) {
// determines the total SStream size by adding up the sizes of all its segments.
long fpos=bin->fpos[key];
tHeader th;
long size=0;
  while (fpos>=0) {
    fseek(bin->tf,fpos,0);
    fread(&th,1,sizeof(th),bin->tf);
    size+=th.size;
    fpos=th.next;
  }
  return size;
}
```

SendSStreamOut

```
void SendSStreamOut(uchar* text,int storageSize,
                    Bin* bin,int key) {
// function to copy an entire SStream to the output file, which
// can happen when there is a large number of identical records.
long fpos=bin->fpos[key];
tHeader th;
  while (fpos>=0) {
    fseek(bin->tf,fpos,0);
    fread(&th,1,sizeof(th),bin->tf);
    while(th.size){
      int textSize=storageSize;
      if (textSize>th.size) textSize=th.size;
      fread(text,1,textSize,bin->tf);
      BufferedWrite(text,textSize);
      th.size-=textSize;
    }
    fpos=th.next;
  }
}
```

RecoverMemory

```
int RecoverMemory() {
// unused top part of the current bin
  return (NUM_KEYS-topKey-1)*sizeof(long);
}
```

FlushBuffer

```
void FlushBuffer() {
int bytes2send=buffer->size-buffer->space;
  if (bytes2send) {
    fwrite(buffer->c,1,bytes2send,theOutFile);
  }
  buffer->space=buffer->size;
}
```

OpenBuffer

```
int OpenBuffer(uchar* storage,int storageSize) {
// allocates 1/8 of available memory, up to a maximum of
// DEF_BUFFER_SIZE as output buffer for outFile

int maxSize=storageSize>>3;
int bufferSize=DEF_BUFFER_SIZE;
  while (bufferSize>maxSize) bufferSize>>=1;
  if (bufferSize>=MIN_BUFFER) {
    buffer=(OutputBuffer*)(storage+storageSize-
      (bufferSize+sizeof(OutputBuffer)-4));
    buffer->size=bufferSize;
    buffer->space=bufferSize;
    return bufferSize+sizeof(OutputBuffer)-4;
  } else return 0;
}
```

The professional web page development tool

- Tables
- Forms
- Page Colors
- Frames
- Text and styled text translation
- Add your own tags
- Preview with different web browsers

NEW VERSION 2.0

For a *fully* functioning demo, point your web browser to… http://webster.northnet.org

Miracle Software Inc. (315) 265-0930


CloseBuffer

```
void CloseBuffer() {
  FlushBuffer();
  buffer=0;
}
```

BufferedWrite

```
void BufferedWrite(uchar* text,int size) {
  if (buffer) {
    if (size > buffer->space) {
      int n1=buffer->space;
      int n2=size-buffer->space;
      memcpy(buffer->c+buffer->size-buffer->space,text,n1);
      buffer->space=0;
      FlushBuffer();
      BufferedWrite(text+n1,n2);
    } else {
      memcpy(buffer->c+buffer->size-buffer->space,
             text,size);
      buffer->space-=size;
    }
  } else {
    fwrite(text,1,size,theOutFile);
  }
}
```

BufferedSend

```
void BufferedSend(Node* node) {
  if (buffer) {
    do {
      BufferedWrite(node->ref,node->len);
      node=node->next;
    } while(node);
  } else Send(node,theOutFile);
}
```

# Distribute Your Software Electronically and Get Paid!

**PACE Software authorization systems let you:**

- Create fully functional time- or launch-limited "trialware" without changing your source code
- Distribute via the Internet, CD-ROM, BBS, floppies and more
- Unlock software with flexibility and security
- Add registration-capture to your software without writing code
- Integrate registration and authorization with popular web servers

**To sample the customer experience, download our demo from www.paceap.com. It's easy, and it's free!**

Want to try out our system on your software? Call 408/297-7444 or send email to sales@paceap.com to order our evaluation kit.

P A C E
ANTI-PIRACY

1082 Glen Echo Ave
San Jose, CA 95125
vox: (408) 297-7444
fax: (408) 297-7441
email: info@paceap.com
web: www.paceap.com

---

# We've Got the Right Tools For Your Web Site!

And our skills are pretty sharp, too.

We can trim your content, hone your design, and implement a Web site that's...well...a cut above the rest.

With today's chopped budgets and pared-down schedules, you need JointSolutions Marketing. We have the experience, and the tools, to make sure your World Wide Web pages aren't...you know...dull.

We even have a Web server, so site maintenance and updates don't slice into your work time.

Call us today and we'll take a whack at your Web needs.

JOINT SOLUTIONS MARKETING

JointSolutions Marketing
Tel: (408) 471-1500
E-mail: info@jointsolutions.com
http://www.jointsolutions.com

*By Jeremy Roschelle*

# Updating Parts Dynamically With SOM

## *Fast, automatic inter-part communication*

As a developer of learning technologies, I am concerned with integrated tools for math and science education. Learners need a wide variety of views, such as meters, graphs, tables, calculators, simulations, data analyzers. Ideally the same object (an accelerated particle, say) can be simultaneously updated in every view.

In the past, all these visualizations would have to be built as a parts of a single monolithic application – a huge effort. With OpenDoc, my colleagues and I are building each view as a separate OpenDoc component. Students can simply drag and drop a particle from a simulation to a graph to a meter. For example, Figure 1 shows a simulation of an oscillating ball, with a vector meter (written by Arni McKinley of Minds in Motion), an adjustable equation (which I wrote), a text editor part, and a graph (by Jim Correia).

The simulation, equation, graph, and vector meter are each separate components, written and compiled at different times, and with different frameworks (PowerPlant, OpenDoc Development Framework, and SamplePart). Yet, as the particle moves, all views instantly update to show its current position, velocity, and acceleration. Furthermore, new views can be added to the system by any developer. For example, Steve Beardslee at Tufts University has written a data collection component that drives a particle with real data from a cart moving on a frictionless track.

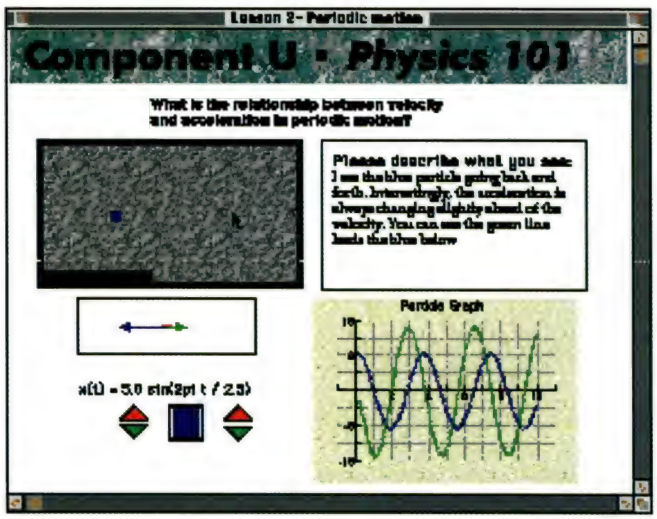
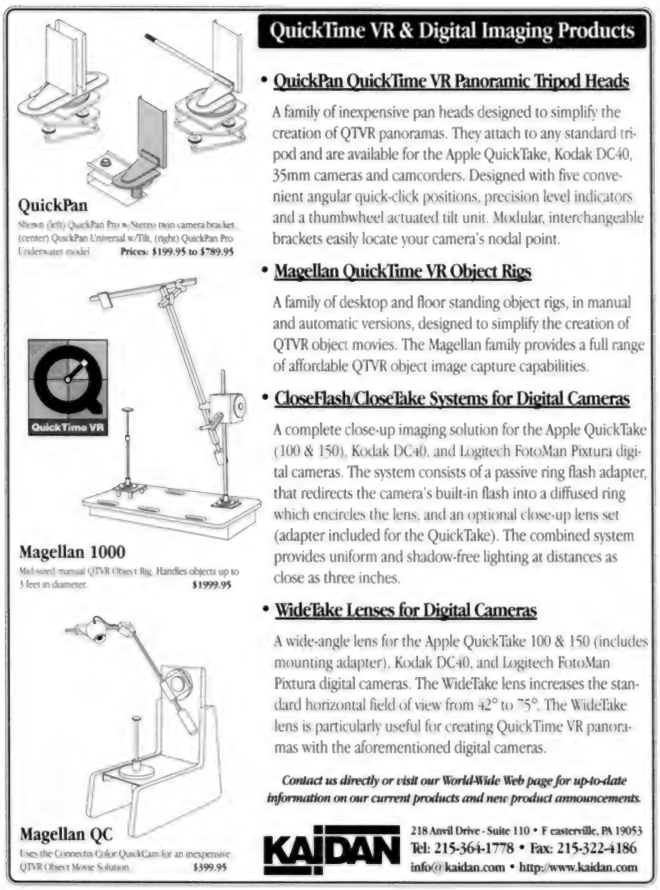



*Figure 1. A component-based physics lesson*

The ability to mix and match components should lead to a rapidly expanding and improving collection of OpenDoc components for math and science education. This article is your opportunity to help education and learn something about OpenDoc at the same time. By the end, you'll know everything you need to contribute a particle viewer to the suite. In fact, you'll build a digital meter that drops right into a window and

***Jeremy Roschelle*** was just a normal San Franciscan: he fixated on his computer's glow, listened to obscure jazz records, voted for all the losing progressive candidates, took his cappucino intravenously, and ate only bagels, burritos and Pad Thai. Then he discovered OpenDoc, and had a vision: powerful components enabling every child to make sense of fundamental math and science concepts. The rest is the future. Contact jeremy at jeremy@soe.berkeley.edu or http://www.slip.net/~jeremy.


QuickTime VR & Digital Imaging Products
QuickPan
Shown (left) QuickPan Pro w/Stereo rotn camera bracket, (center) QuickPan Universal w/Tilt, (right) QuickPan Pro Underwater model
Prices: $199.95 to $789.95
• QuickPan QuickTime VR Panoramic Tripod Heads
A family of inexpensive pan heads designed to simplify the creation of QTVR panoramas. They attach to any standard tripod and are available for the Apple QuickTake, Kodak DC40, 35mm cameras and camcorders. Designed with five convenient angular quick-click positions, precision level indicators and a thumbwheel actuated tilt unit. Modular, interchangeable brackets easily locate your camera's nodal point.
• Magellan QuickTime VR Object Rigs
A family of desktop and floor standing object rigs, in manual and automatic versions, designed to simplify the creation of QTVR object movies. The Magellan family provides a full range of affordable QTVR object image capture capabilities.
QuickTime VR
• CloseFlash/CloseTake Systems for Digital Cameras
A complete close-up imaging solution for the Apple QuickTake (100 & 150), Kodak DC40, and Logitech FotoMan Pixtura digital cameras. The system consists of a passive ring flash adapter, that redirects the camera's built-in flash into a diffused ring which encircles the lens, and an optional close-up lens set (adapter included for the QuickTake). The combined system provides uniform and shadow-free lighting at distances as close as three inches.
Magellan 1000
Mid-sized manual QTVR Object Rig. Handles objects up to 3 feet in diameter.
$1999.95
• WideTake Lenses for Digital Cameras
A wide-angle lens for the Apple QuickTake 100 & 150 (includes mounting adapter), Kodak DC40, and Logitech FotoMan Pixtura digital cameras. The WideTake lens increases the standard horizontal field of view from 42° to 75°. The WideTake lens is particularly useful for creating QuickTime VR panoramas with the aforementioned digital cameras.
Contact us directly or visit our World-Wide Web page for up-to-date information on our current products and new product announcements.
Magellan QC
Uses the Connectix Color QuickCam for an inexpensive QTVR Object Movie Solution.
$399.95
KAIDAN
218 Anvil Drive - Suite 110 • Feasterville, PA 19053
Tel: 215-364-1778 • Fax: 215-322-4186
info@kaidan.com • http://www.kaidan.com

POSITIONS WANTED
Available Immediately
1-800-736-3577
Expert 4D Programmers
Less than 50¢ per hour!
More than 1,000 hours of development went into 4D Toolkit 2.0. With a one time price of $395.00*, it's like hiring a crack team of 4D programmers at less than 50¢ per hour. Call 1-800-736-3577 to order your copy, or to receive a free demo.
4D TOOLKIT™
OPtions Computer Consulting
228 Bleecker Street #19
New York, NY 10014
TEL: 212-645-3577
FAX: 212-633-0336
* upgrade pricing available


interoperates with the simulation. I guarantee you'll have a warm, satisfied feeling when you see your very own component working alongside the ones that Arni, Steve, Jim, and I wrote.

The techniques described in this article also apply to many areas besides education. In fact, they will be useful in any application where one kind of data object is displayed in many different views. Examples would be a personal information manager or a financial checkbook program. If you implement the techniques here, OpenDoc could offer your customers some powerful solutions. They could freely mix and match various viewers to suit their needs. You could sell specific advanced editors to specific niche markets. Fast, object-based data sharing could let your company offer a highly integrated solution with many option packages and upgrade potentials.

### SOM: A Better Way to Share Data

Using the standard OpenDoc APIs, editors and viewers share data by writing out a stream format either to the clipboard or to a link storage unit. This is similar to the familiar techniques for handling the clipboard, or the System 7 publish-and-subscribe mechanism. An obvious problem with stream-based sharing is speed. The delay is particularly painful if the object is large and the client only needs a small portion of the data. A less obvious problem is that you need to standardize a stream format, which can be hard for objects with complex data. Finally, you can only change the data at the source.

A better way to share data is to share an object. Getting data is done with a method call. For example, I can get the position vector of my particle as follows:

```
SDataVector position(3);
myParticle->GetPosition(ev,time,position);
```

The chief advantage of getting data via a method call is speed. A particle might have thousands of data points. Using this method, I get only the data point I need. But there are other advantages as well. Any viewer that has a particle can call its methods to change its data. Furthermore, you can use class inheritance to manage the growing complexity of your objects. For example, my base particle only has data access methods. My sampled particle builds upon this by introducing methods for entering sampled data. My constant acceleration particle adds specific methods for constant acceleration situations. Rather than define one format that meets every need, I can introduce classes that inherit from the core and specialize appropriately for particular situations. Due to the magic of object-oriented programming, older viewers will continue to work with newer, more advanced classes (whereas older viewers would typically crash with newer file formats).

OpenDoc makes it possible to share objects among parts by using the System Object Model (SOM). SOM packages an object binary in a special shared library, where it can be used from many OpenDoc editors, regardless of the compiler or language used to code the editor. Among the numerous advantages of SOM, it solves the fragile base class problem; you

can change an object's interface methods, and older editors will continue to work without your needing to recompile them. While in-depth coverage of SOM is not possible here, you will get a strong feeling for its ease of use and power. Full reference materials for SOM are on the OpenDoc Developer CD and at the OpenDoc Web site.

### Becoming a SOM User

*Soma: an intoxicating plant juice referred to in the literature of Vedic ritual.*

*– Webster's New Collegiate Dictionary, 1972, p. 1356*

Unlike soma, SOM is not a drug. But I'll admit that SOM's ability to make component software flow is somewhat intoxicating, even if it does require a few arcane rituals.

To use a SOM-based class, you first include its header, which has the `.xh` suffix. This is just like including the `.h` header for a C++ class. You call the methods of the class, including `new` and `delete`, just as you would for an ordinary C++ class. You must also include the SOM shared library in your project.

The example below uses the particle class to calculate the position of a ball that is thrown in the air for 10 seconds.

Calculate Position

```
#include "EduConstAccParticle.xh"

double CalculatePosition(Environment *ev)
{
   double finalPos = 0.0;
   EduConstAccParticle *p = nil;
SOM_TRY
   p = new EduConstAccParticle;
   // SDataVector is a C++ utility class that makes it easy
   // to manipulate vectors. 1 is the number of dimensions.
   // SDataVector uses 1-based indexing, because I have trouble thinking
   // of the x-axis as the zeroeth dimension of a 3D vector.
   SDataVector pos(1), vel(1), acc(1);
   pos[1] = 0.0;
   vel[1] = 300.0;
   acc[1] = -9.8;

   // set to constant acceleration motion, with 0.0 initial position and 300.0 m/s
   // and gravitational acceleration.
   p->SetConstantAcceleration(ev,pos,vel,acc);

   // get the value at 10 seconds
   finalPos = p->GetPosition(ev,10.0,1);
   delete p;
SOM_CATCH_ALL
   delete p;
   RERAISE;
SOM_ENDTRY
   return finalPos;
}
```

One difference you will note in using SOM is the first argument, which is always an environment pointer. The environment is a `struct` that SOM uses to return exceptions. The environment is necessary because a SOM object may not be in the same address space as its caller, and thus may not be able to throw an ordinary exception. Upon returning from SOM code you should always check the environment for errors. To make this easy, most SOM headers are produced with an option (`CHECK-ENV`) that automatically converts the environment into a C++ exception if it contains an error. Thus your code can catch SOM-based exceptions and C++-based exceptions with the same try-catch blocks. You should always have at least one exception handler active at the time you call a SOM method (lest you abort your process upon the first bad environment that is returned).

Since you now know how to call SOM methods – it's not very different from C++ – we can move to the real issue: how can particle classes be designed to allow multiple live views?

### Sharing Particles: A Class Act

My goal is to enable multiple components, like simulations, graphs, and tables, to simultaneously display the data of a single object. To accomplish this, I use the model-view-controller strategy (Figure 2), made famous in SmallTalk. In this strategy, the controller (a simulation) makes changes to the model (a particle), and the particle notifies all its views (graphs, meters, tables). The views call the particle's methods to fetch the appropriate data for updating the display.

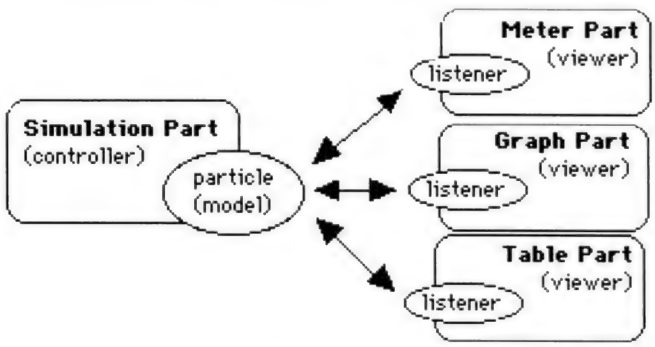



*Figure 2. Model-View-Controller Architecture*

Three classes are needed to implement the strategy in OpenDoc:

1. `EduObject`: a sharable object. (`EduParticle` inherits from `EduObject`.)
2. `EduObjectListener`: registers for change notification with a particular `EduObject`.
3. `EduObjectMgr`: publishes `EduObjects` to other components.

Each display will have a pointer to an `EduObject`. A meter, for example, gets a pointer to an `EduParticle` by receiving a reference in a drag. The method `EduObjectListener::GetObject` uses the particle reference to get access to the `EduParticle`, and the meter uses this `EduParticle` to get data it can display.

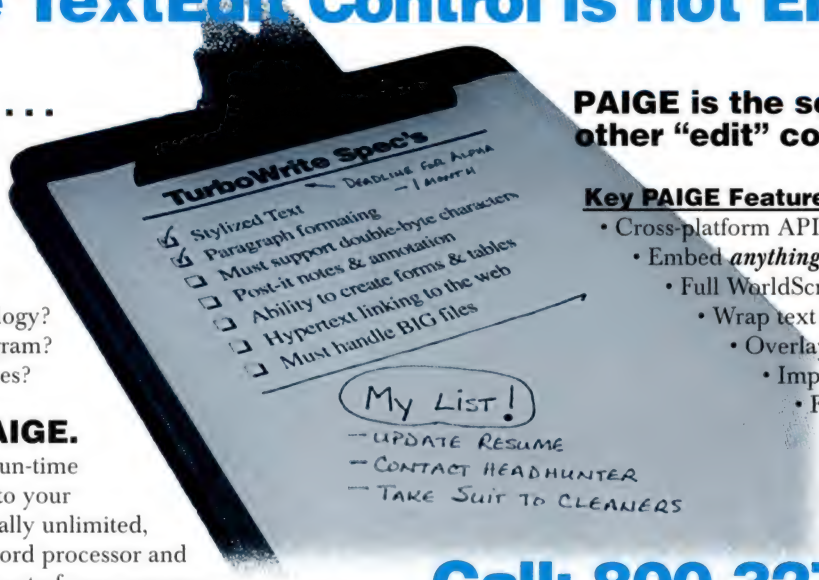
The simulation changes a particle by calling its methods. For example, `SetCurrentTime` changes the simulation time for the particle. This method will also broadcast a message to every `EduObjectListener`. When a meter receives the message in its listener, for example, it will call `EduParticle` methods to extract data and update its display. Every time the

*For C and C++ Developers ....*

# When the TextEdit Control is not Enough.

**Are you writing a . . .**

- Document Viewer?
- Multimedia Product?
- Report Generator?
- Forms Package?
- HTML Editor?
- Email Application?
- Electronic Book Technology?
- Hypertext Enabled Program?
- Program with text features?




**PAIGE is the solution when other "edit" controls fail.**

**Key PAIGE Features**

- Cross-platform API
- Embed ***anything*** into text
- Full WorldScript Support
- Wrap text in/around graphics
- Overlay text on graphics
- Import & Export RTF
- Full MFC support
- Royalty Free

**Then you need PAIGE.**

PAIGE™ is a library of run-time functions you can embed into your application. It will add virtually unlimited, programmable "rich edit" word processor and page layout features to any part of your program.

PAIGE will save you hundreds of precious development hours. Better yet with PAIGE's cross-platform API you can support the most popular desktop operating systems.

**Call: 800-327-6703**

**DataPak Software Inc.**

**Internet:** sales@datapak.com
www.datapak.com/~datapak/mct
**CS:** 76424,3027
**AOL:** DATAPAK1
**Ph:** 360-891-0542
**Fx:** 360-891-0743

**Windows 3.1 • Windows 95 • Windows NT • Macintosh • PowerMacintosh**

simulation changes a particle, all the listeners will be notified and will update their displays.

Finally, it is desirable for the meter to remember its particle when the document is saved, so that the link between the simulation and the meter will persist when the document is opened again. Since the particle is a pointer, it cannot be saved directly. Instead, the meter's listener writes out a reference to the particle (in exactly the same format used to receive a drag). When it is opened, the meter can get the particle object using the `GetObject` method as described above.

The listings below are an abridged version of the interface definitions which specify the API to the classes. These are written in SOM's Interface Definition Language (IDL). IDL reads almost like a C++ class definition, except that each argument is explicitly marked for direction of data flow: `in`, `out`, or `inout`.

EduObject Interface

```
interface EduObject : SOMObject
{
   // abridged interface definition

   // change notification methods
   void RegisterListener(in EduObjectListener listener);
   void UnRegisterListener(in EduObjectListener listener);
   void TellListeners(in long message, inout ODByteArray param);

   // support for dragging: externalizes a reference to this object.
   // storage should be the drag storage unit,
   // EduObjectListener has a method for internalizing this from storage
   void ExternalizeReference(in ODStorageUnit storage);
};
```

EduParticle Interface

```
// inherits from EduObject
interface EduParticle : EduObject
{
   // abridged interface definition

   // inherits all EduObject methods
   // EduConstAccParticle and EduSampledPosParticle
   // add methods for setting data values

   // access to data values
   void GetPosition(in double time, inout EduDataVector vector);
   void GetVelocity(in double time, inout EduDataVector vector);
   void GetAcceleration(in double time, inout EduDataVector vector);

   // time related information
   double GetCurrentTime();
   void SetCurrentTime(in double time);
};
```

EduObjectListener Interface

```
interface EduObjectListener : SOMObject
{
   // abridged interface definition

   // messages are received here
   void ListenToMessage(in long message, inout ODByteArray param);

   // support for receiving drags
   // call HasValidObjectReference to see if you should accept this drag
   // call InternalizeReference to receive the drag
   ODBoolean HasValidObjectReference(
        in ODStorageUnit dragStorage);
   void InternalizeReference(in ODStorageUnit dragStorage);

   // utility methods for storing and retrieving a particle reference
   // in this components storage unit (the OpenDoc generalization of a file)
   void Externalize(
      in ODStorageUnit su,
```

```
    in ODDraftKey key,
    in ODFrame scopeFrame);

  void Internalize(in ODStorageUnit su);


  // get the object associated with this listener
  // may return nil if no object is connected.
  EduObject GetObject();
};
```

EduObjectMgr Interface

```
interface EduObjectMgr : ODExtension
{
  // abridged interface definition

  // called from another component to get an object
  EduObject GetObject(in long id);

  // called from within this component to publish objects
  void AddObject(in EduObject object);
  void RemoveObject(in EduObject object);
};
```

This article takes advantage of the encapsulation offered in these classes by separating the details of *use* from *implementation.* I'll start with how to use the classes, since that is really easy. At the end, I'll draw back the curtain and reveal how it works.

### Building a DigitalMeter Component

The simplest way to make an OpenDoc component is with PartMaker (available at http://www.opendoc.apple.com). PartMaker will generate a complete CodeWarrior project, ready to compile and run. We'll start with SamplePart, a very simple "Hello, World!" type component, and modify it to be a digital meter for a particle. To begin, drop SamplePart C++ onto PartMaker. Name your project "DigitalMeter". You can use whatever company name you like. When PartMaker finishes grinding away, you will have a new folder named DigitalMeter.

Open the project `DigitalMeter.μ`, and perform a make. This will produce a new OpenDoc component. Make an alias to this component and put the alias in your Editors folder. Then drop the component onto the OpenDoc icon (found in `System Folder:Extensions:OpenDoc Libraries:`) to make stationery. The stationery will appear in your Stationery Folder. You should be able to double-click this stationery in order to open your brand new component. Yeah!

Now you should download EduObject and EMSim from my Web site, using the EMDemo link (http://www.slip.net/~jeremy/). Drop the EduObject library, and EMSim and EMMeter parts, into your Editors folder. You will also need ODFDraw or KickStart Draw, as I use these as a generic container. You can get them from the OpenDoc CD or Web site. To see if you have everything installed correctly, try opening the Example document. You should see the simulation and my existing meter. Select the simulation, and choose **Go** from its menu. The simulation will run and the meter will update.

Drag your Digital Meter stationery into the Example document. Your component will be added to the window. Now all we need to do is to make your `DigitalMeter` act like it really is a meter. There are five steps involved:


# DragInstall® 2.0

## Here's proof

## that the more things change

- ▲ New "Quick Build" option lets you build a complete installer in *one easy step.*
- ▲ New features such as locating files and folders, applying patches, and replacing outdated files allow you to build more intelligent installers.
- ▲ Improved support for installing non-archived files simplifies the creation of CD-ROM and network installers.
- ▲ PowerPC-native compression and decompression cuts installation time *in half.*
- ▲ AppleEvent and scripting support allows installations to be automated.

## the more they stay the same

- ▼ Same price–$300 lets you distribute an *unlimited* number of installers for *all* of your products. No yearly fee, no royalties, no hidden costs.
- ▼ Same drag-and-drop interface familiar to all Macintosh users.
- ▼ Same reliability and robustness that developers worldwide have relied on since 1991.
- ▼ Same support policy–free technical support and low (or no) cost upgrades.

For more information or a free demo, call

**1-800-890-9880**

or visit our Web site at

**http://www.sauers.com/draginstall**

Ray Sauers Associates

Ray Sauers Associates
1187 Main Avenue, Suite 1B
Clifton, NJ 07011 USA
voice: 201-478-1970
fax: 201-478-1513
email: info@sauers.com

Join AG Group, Dantz, Neon, Maxum, MR Mac, EFI and others
**Add Pager Notification to YOUR Product**

**with PageNOW!™**

**Imagine the possibilities:**
pro-active error alerts
appointment/to-do/reminders/alarms
database updates
wireless email
voice/fax/email notification

**MacWEEK 3/12/96 ◆◆◆◆ Macworld 3/96 ★★★★/8.8**

Call for free developer kit with sample code (C/C++ and AppleScript).
Ask about our Bundling, Licensing, and Co-Marketing Opportunities!

Expand your products connection, transfer, or emulation options with
**Communications Toolbox Tools**

- ZMODEM file transfer
- PC-ANSI terminal emulation
- TCP/Telnet connection
- Videotex emulation (Prestel, Minitel, BTX)
- Call for others under development
- Custom tool development available
- Available in English, French, German

Also available: Communicate™ Lite: Full Communications Toolbox based Power Mac native terminal program for bundling with modems, ISDN cards, etc.

**Call 1-800-799-1718** mark/space

Tel: (408) 293-7299 Fax: (408) 293-7298 info@markspace.com
Demo software at <http://www.markspace.com>
111 W. Saint John, 4th Floor, San Jose, CA 95113, USA

Copyright ©1996 Mark/Space Softworks. PageNOW!, PageME!, and Communicate are trademarks of Mark/Space Softworks. All other names are trademarks of their respective holders.

1. Creating a particle listener
2. Displaying particle data
3. Receiving a particle drag
4. Listening to particle changes
5. Saving a particle reference

### 1. Creating a particle listener

To begin, add the EduObject library to your project (it is a shared library so it won't add its code into your component). Then add the `SDataVector.cp` file to your project.

The code below creates an `EduObjectListener` in your `DigitalMeter`. You need a listener because it is both your handle to the `EduParticle` (which is created by the simulation part) and your way to receive messages when the particle changes. As you can see, creating a SOM object is no big deal.

DigitalMeter::Initialize

```
void DigitalMeter::Initialize( Environment* ev )
{
  SOM_Trace("DigitalMeter","Initialize");

  fListener = new EduObjectListener;
  fListener->InitEduObjectListener(
    ev,fSelf,ListenToMessage,(unsigned long)this);
  CopyPascalString(fCurrentValue,"\pDrag a particle here");
  // unchanged code below here, omitted for clarity
}
```

DigitalMeter::ReleaseAll

```
void DigitalMeter::ReleaseAll( Environment* ev )
{
  SOM_Trace("DigitalMeter","ReleaseAll");
  delete fListener;
  // unchanged code below here, omitted for clarity
}
```

Below I've listed the additions to the `DigitalMeter` header for all the code in this article:

```
// original top of header omitted for clarity
// my changes below
#include <EduObjectListener.xh> /
#include "ListenerHelper.h"
#include <EduParticle.xh> //+
#include "SDataVector.h"
#include <DgItmIt.xh>
#include <FacetItr.xh>
#include <stdio.h>

class DigitalMeter {
  // new methods
  public:
  static void ListenToMessage(
    Environment *ev, unsigned long inRefCon,
    long inMessage, ODByteArray *ioParam);

  ODBoolean  IsAcceptable(
      Environment *ev, ODDragItemIterator *dragInfo);
  void       ReceiveDrop(
      Environment *ev, ODDragItemIterator *dragInfo);

  void   UpdateDisplay(Environment *ev);
  void   DrawAll(Environment *ev);

  // new members
  private:
  EduObjectListener *fListener;
  Str255            fCurrentValue;
  // unchanged class definition below here, omitted for clarity
};
```

### 2. Digital display

The digital display just draws a string containing the position of the particle. It gets the particle from the listener. (You will see how the listener gets the particle in the next section.) The string is stored in the `fCurrentValue` member, which will be set by the `UpdateDisplay` method. Once the string is built, the `UpdateDisplay` method calls `DrawAll`, which redraws every facet of every frame with the new string.

DigitalMeter::UpdateDisplay

```
void DigitalMeter::UpdateDisplay(Environment *ev)
{
  EduParticle *particle = (EduParticle *)
    fListener->GetObject(ev);
  if (! particle) return;

  // get current position
  double time = particle->GetCurrentTime(ev);
  SDataVectorpos(2);
  particle->GetPosition(ev,time,pos);

  // use ANSI routines to write numbers into Pascal string
  char *output = (char *)&fCurrentValue[1];
  sprintf(output, "%.1f,%.1f",pos[1],pos[2]);
  // set length of Pascal string
  fCurrentValue[0] = strlen(output);

  DrawAll(ev);
}
```

# Need to add plotting capabilities to your program?

Save time and money with one of our three popular Superplot libraries

SuperPlot™

**Plotting/Graphics Libraries**

**From $150 - $250**

- *Choose SuperPlot, SuperPlot+, or SuperPlotPRO*
- *PowerMac version available*
- *Works with Fortran, C or Pascal*
- *Data analysis*
- *Batch processing*
- *19 plot types*
- *Windows 95/NT version available for 2D and 3D*

For free demo and info contact **SuperSoft**

498 E. Robin Road • Orem, UT 84057 • Voice (801) 225-4356 • Fax (801) 226-6276
http://www.SuperSoft.com


DigitalMeter::DrawFrameView

```
void DigitalMeter::DrawFrameView(
        Environment* ev,ODFacet* facet )
{
  // setup text characteristics
  TextSize(12);
  TextFont(1);
  TextFace(bold);
  PenNormal();
  FontInfo finfo;
  GetFontInfo(&finfo);
  MoveTo(0,finfo.ascent + 2);

  EraseRect(&port->portRect);
  DrawString(fCurrentValue);
}
```

DigitalMeter::DrawAll

```
void DigitalMeter::DrawAll(Environment *ev)
{
  // iterate through frames, then facets
  CListIterator fiter(fDisplayFrames);
  for ( CFrameProxy* proxy = (CFrameProxy*) fiter.First();
        fiter.IsNotComplete();
        proxy = (CFrameProxy*) fiter.Next() )
  {
    // draw only frames in RAM, not any still on disk
    if (proxy->FrameIsLoaded(ev)) {
      ODFrame  *frame = proxy->GetFrame(ev);
      // iterate through facets
      ODFrameFacetIterator *iter =
        frame->CreateFacetIterator(ev);

      for (ODFacet *facet = iter->First(ev);
         iter->IsNotComplete(ev);
         facet = iter->Next(ev)) {

        Draw(ev,facet,kODNULL);

      }
      delete iter;
    }
  }
}
```

### 3. Its a drag!

The `DigitalMeter`'s listener initially learns about a particle by receiving a dragged particle reference. To receive any drag, each frame must have its flag set correctly. We set the flag when your frame is added or connected:

DigitalMeter::DisplayFrameAdded

```
void DigitalMeter::DisplayFrameAdded(
        Environment* ev, ODFrame* frame )
{
  SOM_Trace("DigitalMeter","DisplayFrameAdded");
  frame->SetDroppable(ev,kODTrue);
    // unchanged code below here, omitted for clarity
}
```

DigitalMeter::DisplayFrameConnected

```
void DigitalMeter:: DisplayFrameConnected (
         Environment* ev, ODFrame* frame )
{
  SOM_Trace("DigitalMeter","DisplayFrameConnected");
  frame->SetDroppable(ev,kODTrue);
    // unchanged code below here, omitted for clarity
}
```

The basic principle of receiving a dragged particle is simple. When something is dragged into your frame, OpenDoc will call `DragEnter` and pass in a storage unit that contains the dragged content. If your part decides the drag is acceptable, then when the user releases the drag, your receive drop method will be called. `EduObjectListener` has two methods for receiving drags. `HasValidObjectReference` returns `true` if the listener can accept the drag, and `InternalizeObjectReference` actually handles the drop.

DigitalMeter::IsAcceptable

```
ODBoolean  DigitalMeter::IsAcceptable(
        Environment *ev, ODDragItemIterator *dragInfo)
{
  ODStorageUnit *sue = dragInfo->First(ev);
  return fListener->HasValidObjectReference(ev,sue);
}
```

DigitalMeter::ReceiveDrop

```
void   DigitalMeter::ReceiveDrop(
         Environment *ev, ODDragItemIterator *dragInfo)
{
  ODStorageUnit *sue = dragInfo->First(ev);
  fListener->InternalizeReference(ev,sue);
  SetDirty(ev);
  UpdateDisplay(ev);
}
```

# Can you spot the difference?

Plenty of people can't. Because whether you update software with a full set of program disks, or a file made with UpdateMaker 2, the result is the same. Guaranteed. UpdateMaker updates are totally ***reliable***. Its system of 32-bit checksums ensures that it updates the right file.

And UpdateMaker is ***easy-to-use*** – simply specify the files and UpdateMaker builds the update. There is no scripting or use of ResEdit. It's even easier for end-users – just one button to press.

UpdateMaker 2 works with any type of Macintosh file. The updater files are extremely ***compact***. And the program options numerous. You can preserve or override user customisations. Save files in Binhex format. Update up to 20 old versions with one file.

**The real difference is the savings in time and money. Which explains why many of the best-known names in software development have been using UpdateMaker for years.**

**Distribution is unrestricted and royalty-free.**

Accelerated for Power Macintosh

**Only $225, order now by fax: (415) 964 2886**

**Email: update@adi.com.au**

# UpdateMaker 2™

ADInstruments, 2225 Grant Road, Suite #4, Los Altos, CA 94024 Phone: (415) 964 2878

## ALL BUGS ARE STUPID.

But spending tedious hours trying to track them down is dumber still. Why not let a tool do the work? QC can find many of those mistakes automatically. Ever write data beyond the end of a memory block? Ever rely on a handle that was purged? Ever call DisposeHandle on a resource or ReleaseResource on a handle? Sure you have! Maybe you just haven't found out about it yet... QC finds these errors and more.

**BECAUSE:**

Every programmer makes mistakes.
All programs ship with bugs.
Marketing just cut the beta.
You could use some sleep.

**YOU NEED:**

QC

QC is cool and, unlike other development tools, QC is easy. Try it **FREE:**

1. Connect to our web site
2. Download QC (less than 200K)
3. Send us email to get a serial #
4. Run the installer
5. Run your program
6. Press shift-option-q

*"I only have 6 non-Apple Control Panels on my development machine. QC is one of them. 'Nuff said."*
*-Bill Goodman, Compact Pro author*

*"We wouldn't ship a product without QC's approval."*
*-Mate Gross, Claris Corporation*

**NOW POWERPC NATIVE ! EXISTING USERS UPGRADE FREE!**

Onyx

**$99**

Onyx Technology 7811 27th Ave Bradenton, FL 34209
Tel: 941.795.7801 Fax: 941.795.5901
Web: http://www.std.com/onyxtech/
AOL: OnyxTech AppleLink: D2238 CIS: 70550,1377

Unfortunately, `SamplePart` has only stub methods for receiving drops, so we have to delve into the SOM class implementation of the part. The code below looks funny, but is basically just a C++ function implementation with a little extra work to interface with SOM. As with all methods in the SOM class implementation, this code just sets up exception handling and then delegates the work to the `DigitalMeter` class. For simplicity's sake, drag hiliting is not implemented here.

som_DigitalMeter__DragEnter

```
SOM_Scope ODDragResult SOMLINK som_DigitalMeter__DragEnter(
        PartCo_som_DigitalMeter *somSelf, Environment *ev,
        ODDragItemIterator* dragInfo, ODFacet* facet,
        ODPoint* where)
{
  PartCo_som_DigitalMeterData *somThis =
    PartCo_som_DigitalMeterGetData(somSelf);

  ODBoolean  result = kODFalse;
  SOM_TRY
    result = _fPart->IsAcceptable(ev,dragInfo);
  SOM_CATCH_ALL
  SOM_ENDTRY
  return result;
}
```

som_DigitalMeter__DragWithin

```
SOM_Scope ODDragResult  SOMLINK som_DigitalMeter__DragWithin(
        PartCo_som_DigitalMeter *somSelf, Environment *ev,
        ODDragItemIterator* dragInfo, ODFacet* facet,
        ODPoint* where)
{
  // same as DragEnter
  return somSelf->DragEnter(ev,dragInfo,facet,where);
}
```

som_DigitalMeter__Drop

```
SOM_Scope ODDropResult  SOMLINK som_DigitalMeter__Drop(
        PartCo_som_DigitalMeter *somSelf, Environment *ev,
        ODDragItemIterator* dropInfo, ODFacet* facet,
        ODPoint* where)
{
  PartCo_som_DigitalMeterData *somThis =
    PartCo_som_DigitalMeterGetData(somSelf);

  SOM_TRY
    _fPart->ReceiveDrop(ev,dropInfo);
  SOM_CATCH_ALL
  SOM_ENDTRY
    return kODDropCopy;
}
```

### 4. Listening in

Now we get to the easy part. Every time the particle changes, your `DigitalMeter` will get a message. The `ListenToMessage` method simply calls `UpdateDisplay` to display the new values

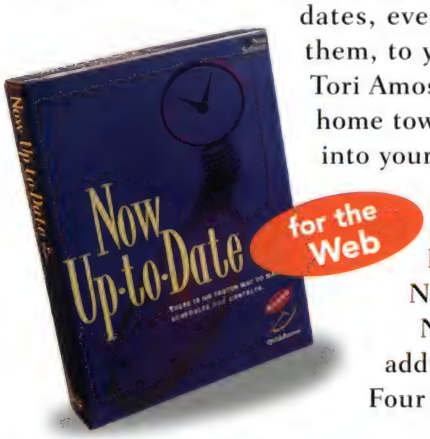
DigitalMeter::ListenToMessage

```
void
DigitalMeter::ListenToMessage(
        Environment *ev, unsigned long refCon,
        long inMessage, ODByteArray *ioParam)
{
  DigitalMeter *me = (DigitalMeter *)refCon;
  switch (inMessage) {
    case EduParticle_msgDataChanged:
    case EduParticle_msgTimeChanged:
      me->UpdateDisplay(ev);
      break;
    default:
      break;
  }
}
```

# What kind of person can get Hootie & The Blowfish, Tori Amos, *and* Collective Soul on his calendar?

You. And no recording industry experience required. Announcing Now Up-to-Date,™ the only schedule and contact manager that brings you the world of the Web.

Take off to sites like Atlantic Records. With a single drag and drop, you can add your favorite artist's concert dates, even a whole group of them, to your calendar. Find Tori Amos' tour date for your home town and simply drag it into your calendar, along with a hot link back to the Atlantic site. No typing. No extra steps.




Need someone's e-mail address? Surf over to the Four11 site, where you can add it into your Now Up-to-Date address book in a single drag and drop operation.

Got an intranet at work? Now Up-to-Date Web Publisher lets you post calendars and address listings with absolutely no HTML programming. Everyone has easy access to employee directories, client lists and project schedules through their Web browser—Windows 95, 3.1 and Mac OS alike.

To download a free trial of Now Up-to-Date, visit the Now Software Time Management Network at http://www.nowsoft.com/trial9. You'll find everything you need to get started, including links to the growing list of Web sites already publishing calendars and address books.

To find out how Now Up-to-Date can get you connected, call 1-800-714-6286.

Sunglasses optional.

Mac OS

Designed for Microsoft® Windows®95

ATLANTIC

1 *Visit www.atlantic-records.com*
*Now drag-and-drop*

2

3 *any event into your personal calendar.*

Now Up-to-Date, the Now Software logo, and "It's about time." are trademarks of Now Software, Inc.; Windows is a trademark of Microsoft Corp.; Macintosh and Mac OS are trademarks of Apple Computer, Inc.; the Atlantic Records logo is a trademark of Atlantic Recording Corporation; Four11 is a trademark of Four11 Corp.

# Help Wanted

Join the best virtual engineering team in the industry! The Now Utilities™ Team needs engineers and utilities for:

- **Now Utilities 7.0**
  *We need new utilities and engineers to enhance existing utilities*
- **Now Utilties for Copland**
  *We need new utilities for Copland*
- **Now Utilities Plug Ins™**
  *We need new Plug Ins for Now Utilities 6.0, 7.0 and Now Utilities for Copland*

Join our virtual engineering team and we'll give you the exposure, recognition and royalties only Now Utilities distribution can provide.

Now Software

http://www.nowsoft.com

For more information on becoming part of the Now Utilities Team send mail to: utilities@nowsoft.com.



# Apprentice 4

The Definitive Collection of Source Code and Utilities for Mac Programmers

Over 640 megabytes of high-quality and up-to-date Mac-only source code examples. **All of the source code examples are new and updated in this release!** Apprentice source is mostly C, C++, and Pascal, using CodeWarrior, Symantec, and MPW. Includes examples of applications, games, code resources, control panels, system extensions, plug-in modules, hundreds of snippets, and much more!

**$35. Upgrade from any previous Apprentice release for only $25!**

Shipping included for U.S. and Canadian orders. Add $5 for shipping outside the U.S. and Canada.

**Visa, MasterCard, American Express, and Discover gladly accepted**

**Celestin Company, Inc., 5652 NE Meadow Road, Kingston, WA 98346**
**800 835 5514 • 360 297 8091 • 360 297 8092 fax**
**Internet: celestin@celestin.com • http://www.celestin.com/**


### 5. Being persistent

As a final touch, it's nice to store the particle reference in your part's content, so that the same particle can be displayed in the meter when the document is opened the next time. `EduObjectListener` handles the work of doing this for you.

DigitalMeter::InternalizeContent

```
void
DigitalMeter::InternalizeContent(
        Environment* ev, ODStorageUnit* storageUnit )
{
  if (ODSUExistsThenFocus(ev,storageUnit,
        kODPropContents,  kDigitalMeterKind )) {
      fListener->Internalize(ev,storageUnit);
      UpdateDisplay(ev);
  }
}
```

DigitalMeter::ExternalizeContent

```
void
DigitalMeter::ExternalizeContent(
        Environment* ev, ODStorageUnit*  storageUnit,
        ODDraftKey key, ODFrame* scopeFrame)
{
  ODSUForceFocus(ev, storageUnit,
    kODPropContents,kDigitalMeterKind);
  fListener->Externalize(ev,storageUnit,key, scopeFrame);
}
```

After entering these changes, make your part and reopen the Example document. Try to drag a particle into your digital meter. You should begin seeing live updates!

### How It Works: A Look Behind the Scenes

You might by now be wondering how `EduObject` and its related classes do their work. The key magic is really provided by SOM. Each `EduObject` class implementation is compiled into the EduObject library. Each component dynamically links to the library at run time. This enables each component to share the same `EduObject` code for particles and listeners.

To implement the classes, I first wrote the interface definitions (`IDL` files). I then submitted these to the SOM compiler, which emits `.xh`, `.xih`, and `.cpp` files. The `.cpp` file has method stubs, which I filled in with C++ implementations of each method. I compiled these into the shared library and exported each SOM class data structure.

If you want to explore the class implementations in depth, you can read the full sources, which you received when you downloaded EduObject from my Web page. Here I can only provide a quick tour of some of the key concepts behind dragging, persistence, and change notification.

To perform a drag in OpenDoc, you write data into a storage unit that represents the content of the drag. The storage unit can contain multiple flavors of data, each in a separate value. `EduObject` has an `ExternalizeReference` method which writes down two pieces of information, the ID of its containing part and the ID of the particle within the part.

EduObjectExternalizeReference

```
SOM_Scope void  SOMLINK EduObjectExternalizeReference(
        EduObject *somSelf, Environment *ev,
        ODStorageUnit* storage)
```

```
{
  EduObjectData *somThis = EduObjectGetData(somSelf);
SOM_TRY
  ODSUForceFocus(ev,storage,
        kODPropContents,EduObject_ReferenceKind);
  // write partID
  ODPart *part = (ODPart *)_fMgr->GetBase(ev);
  long partID = part->GetID(ev);
  StorageUnitSetValue(storage,ev,sizeof(long), &partID);
  // write objectID
  long objectID = somSelf->GetID(ev);
  StorageUnitSetValue(storage,ev,sizeof(long), &objectID);
SOM_CATCH_ALL
SOM_ENDTRY
}
```

To receive the drag, `EduObjectListener::HasValidObjectReference` just checks to see that the correct value type (flavor) is available in the drag. `EduObjectListener::InternalizeObjectReference` retrieves the part ID and the object ID. (Persistence basically works the same way: `Externalize` writes down a part ID and object ID, and `Internalize` reads these numbers back in.)

EduObjectListenerHasValidObjectReference

```
SOM_Scope ODBoolean  SOMLINK EduObjectListenerHasValidObjectReference(
        EduObjectListener *somSelf, Environment *ev,
        ODStorageUnit* dragStorage)
{
  EduObjectListenerData *somThis =
    EduObjectListenerGetData(somSelf);

  ODBoolean  result = kODFalse;

SOM_TRY
  result = dragStorage->Exists(ev,
      kODPropContents,EduObject_ReferenceKind,0);
SOM_CATCH_ALL
SOM_ENDTRY

  return result;
}
```

EduObjectListenerInternalizeReference

```
SOM_Scope void  SOMLINK EduObjectListenerInternalizeReference(
        EduObjectListener *somSelf, Environment *ev,
        ODStorageUnit* dragStorage)
{
  EduObjectListenerData *somThis =
    EduObjectListenerGetData(somSelf);

SOM_TRY
  if (ODSUExistsThenFocus(ev,dragStorage,
        kODPropContents,EduObject_ReferenceKind)) {
    StorageUnitGetValue(dragStorage,ev,sizeof(long),&_fPartID);
    StorageUnitGetValue(dragStorage,ev,sizeof(long),&_fObjectID);
SOM_CATCH_ALL
SOM_ENDTRY
}
```

Now how do we actually get a particle from the simulation to the meter? There is no way to directly get a particle (or any other object) out of an OpenDoc part; the `ODPart` interface simply does not have a method for returning particles! Therefore it is necessary to extend `ODPart` with an `ODExtension`. `EduObjectMgr` is an extension that adds a specific method for returning the object with a specific ID. This extension effectively publishes a set of particles so that other components can see them.

Within your meter, `EduObjectListener::GetObject` works as follows: First it uses the part ID to retrieve the


CRUSHER!™ Data Compression Toolkits

The best high performance, portable compression libraries for DOS, Windows, OS/2, Unix, Macintosh, embedded systems, and practically anything else, period.

| | | |
|---|---|---|
| Robust, 45-Function API | DOS | $249 |
| Buffer and File Compression | Win16 | $299 |
| Portable Archives and Data | Win32 | $299 |
| Disk Spanning | OS/2 | $349 |
| Encryption | Unix | $349 |
| Self-Extracting Executables | Macintosh | $349 |
| On-line Help | | |
| Full C Source Code | | |

FREE DEMO

DC Micro Development

Tel 606-268-1559
Fax 606-266-0726
info@dcmicro.com
http://www.dcmicro.com

Call 1-800-775-1073

Adobe

Developer Tools to Support Adobe® Technologies

Adobe provides a complete set of tools and services for your development needs. Whether you want to integrate Adobe Acrobat® capabilities into your applications, add PostScript® language support to your products or create powerful Graphics Application Plug-ins, Adobe has the tools.

These Software Development Kits now available:

- Adobe Acrobat Plug-ins
- Adobe PostScript Language
- Adobe Photoshop®
- Adobe Illustrator®
- Adobe Premiere®
- Adobe PageMaker® & FrameMaker®

To have information faxed to you, call (206) 628-5737 and request document 1220. Or visit our web page: http://www.adobe.com/

Adobe Developers Association
1585 Charleston Road, Mt. View, CA 94039

Features | Benefits

Create a *working* GUI element with just 1 line of code!

Tools Plus 3.1

Get extraordinary results using ordinary C, C++ or Pascal. Only 170 core routines (about 370 total) replace the need for thousands of Toolbox routines, and thousands of lines of code. Tools Plus has you covered with…

- **Windows** (all kinds, auto-positioning)
- **Cursors** (color, animation, auto-change)
- **Scroll bars** (speed control, live scrolling)
- **Picture buttons** (the best anywhere)
- **Makes CDEFs & LDEFs work automatically**
- **Panels** (3D, flat, many options)
- **Menus** (pull-down and hierarchical)
- **Editing fields** (w/scroll bars and filters)
- **Edit menu** (auto-editing with undo/redo)
- **Full color and multi-monitor support**
- **World-class custom buttons and sliders**
- **Tool bar**
- **Floating palettes**
- **Buttons** (all kinds)
- **List boxes**
- **Pop-up menus**
- **Clipboard**
- **Dynamic alerts** (no resources req'd)
- **Automatic event handling**
- **and much more!**

One line creates it. One line makes it work.

SuperCDEFs™

World-Class Controls for the Discerning Developer

For all Macintoshes and system versions

Full color and multi-monitor support

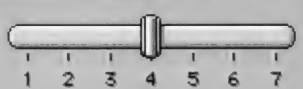
680x0, PowerMac and Fat/Safe format

The finishing touch for a complete 3D look

- ✔ 7 buttons, 9 tabs and 11 sliders (flat/3D text, flat/3D body, raised/inset title, soft/bold shadows, variable shapes) plus a thermometer
- ✔ Customizable check boxes plus undefined state
- ✔ Sliders options: tick marks (1 side/2 sides/none), scale (forward/reversed/none), snap or step to mouse, smart scaling

only $89*

*Free with purchase of Tools Plus Developer Kit




Water's Edge Software

2441 Lakeshore Road West
Box 70022
Oakville, Ontario
Canada L6L 6M9
Phone: (416) 219-5628
Fax: (905) 847-1638

e-mail: WaterEdgSW@aol.com
http://www.interlog.com/~wateredg

Tools Plus for
Symantec (THINK) C/C++ $149
THINK Pascal $149
THINK C/C++ & Pascal $199
CodeWarrior Bronze $199
CodeWarrior Gold $249

(We accept VISA and Amex. Add $10 for shipping.)
*Call for Academic pricing

Free Evaluation Kit: Available on AOL, CIS, Internet and at our web site

MW ★★★★
Macworld, Feb'96

`ODPart` that contains the particle. Then it gets the `EduObjectMgr` extension from that part. Finally, it asks the extension to return the particle with the specified object ID.

EduObjectListenerGetObject

```
SOM_Scope EduObject*  SOMLINK EduObjectListenerGetObject(
        EduObjectListener *somSelf, Environment *ev)
{
  EduObjectListenerData *somThis =
    EduObjectListenerGetData(somSelf);
  // return immediately if it is in the cache
  if (_fObject) return _fObject;

  ODPart *part = kODNULL;
  EduObjectMgr *ex = kODNULL;
SOM_TRY
  // To get the particle, we first get its owner part
  part = _fThisPart->GetStorageUnit(ev)->
    GetDraft(ev)->AcquirePart(ev,_fPartID);

  ODISOStr extName = EduObjectMgr_ExtensionName;
  if (part!= kODNULL && part->HasExtension(ev,extName)) {
    // now get the extension of the part, and use it to get the object
    ex = (EduObjectMgr*)part->AcquireExtension(ev,extName);
    _fObject = ex->GetObject(ev,_fObjectID);
    ex->Release(ev);
    ex = kODNULL;
  }
  part->Release(ev);
  part = kODNULL;

  // register to receive change notification
  if (_fObject)
    _fObject->RegisterListener(ev,somSelf);

SOM_CATCH_ALL
  if (part) part->Release(ev);
  if (ex) ex->Release(ev);
SOM_ENDTRY
  return _fObject;
}
```

When a `EduObjectListener` has an `EduObject`, it calls `EduObject::RegisterListener`. The `EduObject` simply stores all registered listeners in a list. `EduObject::TellListeners` will transmit a message to every listener in the list. For example, when `SetCurrentTime` is called for a particle, it in turn calls `TellListeners` with `msg_TimeChanged`.

EduParticleSetCurrentTime

```
SOM_Scope void  SOMLINK EduParticleSetCurrentTime(
        EduParticle *somSelf, Environment *ev, double time)
{
  EduParticleData *somThis = EduParticleGetData(somSelf);
  _fTime = time;

  SOM_TRY
    ODByteArrayba;
    ba._length = ba._maximum  = 0;
    ba._buffer = kODNULL;
    somSelf->TellListeners(ev,msgTimeChanged,&ba);
  SOM_CATCH_ALL
  SOM_ENDTRY
}
```

EduObjectTellListeners

```
SOM_Scope void  SOMLINK EduObjectTellListeners(
        EduObject *somSelf, Environment *ev,
        long message, ODByteArray* param)
{
  EduObjectData *somThis = EduObjectGetData(somSelf);
  if (! _fIsTelling) return;
```

```
SOM_TRY
  ListenerIter iter(_fListenerList);
  for(   EduObjectListener *thisListener = iter.First();
         iter.IsNotComplete();
         thisListener = iter.Next()) {
    thisListener->ListenToMessage(ev,message,param);
  }
SOM_CATCH_ALL
SOM_ENDTRY
}
```

### Conclusion: Towards Fast, Flexible Suites

The model-view-controller technique used here to implement fast, object-based data sharing is a common technique within many monolithic applications. It is powerful because it separates the data model (particle) from its interface (viewer components), while providing change notification techniques (listeners) to keep everything nicely synchronized. It allows a suite of editors and viewers all to operate and display the same data.

Before OpenDoc, there was no way to use this technique outside the bounds of a monolithic application. SOM, in particular, makes it possible to share objects across separately authored and compiled components. SOM makes it possible to distribute an object library and the object interfaces for a specific data type. Then anybody can write a component to view and edit that type of data. (And in the future, with DSOM, objects can even be distributed across the Internet).

OpenDoc and SOM provide a powerful way to enable a group of small developers to collectively produce a software suite. For example, four projects have already contributed educational components based on `EduObject`. Together we are quickly building a powerful suite for physics and calculus, each contributing based on our particular strengths. The components in this suite integrate smoothly, and will provide teachers much flexibility in customizing their presentation to their students.

(You can contribute an educational part too! How about a nice analog clock, or a strip chart, or a speedometer? If you feel inspired, I'd love to see a copy of the result.)

Notice, however, that the specifics of particles are completely separated from the general interfaces that enable data sharing. You could use `EduObject`, `EduObjectListener`, and `EduObjectMgr` with any kind of object – a financial transaction, a personal contact record, or a calendar entry. To do so, just define a new class that inherits from `EduObject`.

Moreover, there is no reason to limit this power to groups of small developers who are building a suite. A larger developer that produces a deep and rich collection of editors can benefit as well. If I were building a Personal Information Manager today, I would build each contact entry as an object, and make each possible view (e.g. calendar, address book, to-do list) be a separate component. The obvious benefit is that then I could provide customers with custom solutions by substituting components. Some might like a more powerful calendar, whereas others have specific address-book requirements. Moreover, the larger developer could use components to manage the upgrade process; each component


Macintosh® Common Lisp — Digitool

An Object-oriented Dynamic Language

**Now PowerPC Native!**

For the full story visit our Web site:
**http://www.digitool.com/mcl5.html**

Netscape: Digitool, Inc. Home Page
Back Forward Home Reload Images Open Print Find Stop
Location: http://www.digitool.com/mcl5.html
Digitool, Inc.
Welcome and thank you for stopping by.
Our expertise is Macintosh and Common Lisp development. Our main product is MCL (Macintosh Common Lisp). Digitool was formed over a year ago to acquire MCL from Apple Computer and to bring you a PowerPC native implementation. Most recently, we have also been contracted by Apple to port the Apple Dylan Technology Release to MCL for the PowerPC. Our development effort is sponsored by a number of MCL customers and our group includes key members of the Apple and Coral MCL development teams.
If you just happened here, then you've stumbled across the MCL Phenomenon, one of the best kept secrets in Macintosh development.
News Flash!
December 15, 1995. Digitool, Inc. to port Apple's Dylan Technology Release to PowerPC MCL! Check out the full text of the announcement.
Check out the earlier news flashes!

Digitool, Inc. • One Main Street • Cambridge, MA 02142
tel: (617) 441-5000 • fax: (617) 576-7680 • email: info@digitool.com

**Develop OpenDoc Parts Lightning Fast!**

Developing with OpenDoc means learning a new software architecture. But by taking a Developer University class and using the OpenDoc Development Framework (ODF) you'll come up to speed quickly and reduce your programming time.

This hands-on, five-day class—Creating Open Doc Parts—will build your OpenDoc Framework programming skills. Knowledgeable instructors will help you focus on the most important components and steer you away from common pitfalls.

**1996 Course Offerings:**

❒ July 15-19 ❒ Sept. 16-20
❒ Oct. 28-Nov 1 ❒ Dec. 9-13

**Price: $1500**

For more information, contact Apple Developer University at DEVUNIV.applelink.apple.com or visit our web site at: **http://dev.info.apple.com/du.html.**

# MacRegistry™

## Developer Job Opportunities

If you are a Macintosh developer, you should register with us! We have a database that enables us to let you know about job opportunities. When we are asked to do a search by a client company the database is the first place we go. There is no charge for registering. The database service is free. Geographic Coverage is nationwide.

Marketability Assessment - To get a specific feel for your marketability send a resumé via Email or call. You may also request a Resume Workbook & Career Planner.

Discreet - We are very careful to protect the confidentiality of a currently employed developer.

Scientific Placement is managed by graduate engineers, we enjoy a reputation for competent & professional job placement services and we are Mac fanatics.

**1-800-231-5920 • das@spi.com • Fax 1-800-757-9003**
**http://www.scientific.com**

## Scientific Placement, Inc.

MT. Box 19949, Houston, TX 77224, 713-496-6100 Fax: 713-496-0373
MT. Box 71, San Ramon, CA 94583 510-733-6168 beth@spica.bdt.com
MT. Box 202676, Austin, TX 78720-2676 512-260-0123 lej@zilker.net
AppleLink: D1580; Compuserve: 71250,3001; AOL: davesmall



THE TRATTNER NETWORK

COMPUTER CONSULTING SERVICES

The Trattner Network (TTN) has the best opportunities for Macintosh developers in Northern California and Nationwide.

TTN represents clients with projects in Internet, Multimedia, Codewarrior and OpenDoc development among others.

Now you can visit us our web site at www.tratnet.com and see what openings are available for....

- **Software Developers**
- **Hardware/Firmware Engineers**
- **QA/QC Professionals**
- **Project Coordinator/Managers**
- **Multimedia Developers**
- **Network Professionals**

**The Trattner Network** has a unique history in **Mac consulting** coupled with exposure to emerging technologies. If you are looking for a chance to enhance your skills and marketability, **please send, fax, e-mail or link your resume to:**

**The Trattner Network**
Attn: Emily Hoolhorst
170 State Street, Suite 240 • Los Altos, CA 94022
Phone: 415•949•9555 ext.115 ; Fax: 415•949•1026
AppleLink: trat.net ; E-mail: emily@tratnet.com
http://www.tratnet.com

THE DIGITAL TALENT SOURCE


# HELP MAKE MACTECH WORK

Here at *MacTech Magazine*, we rely heavily on outside writers for most of the material that appears in our pages. If readers did not participate in the magazine, sending us their ideas and taking the time to write articles, there would be no *MacTech*. *MacTech Magazine* is not a staff of writers sending a constant stream of one-way messages outwards; it's a living, evolving network of readers conversing with one another, educating one another, sharing their knowledge, their experience, their interest, their trials and tribulations and joys and successes in the constantly unfolding story of programming the Macintosh. *MacTech Magazine* doesn't just happen: it's what the community makes it. If we carry reports of future trends and technologies, if we teach useful methods, if we review new books and tools, if we provoke thought, provide help, ride the wave of current interests and concerns, it is only because we reflect the thoughts of our readers, who speak through our pages.

You are invited to involve yourself in this exciting conversation amongst readers. No matter who you are, no matter what your credentials may be, if you have a tale to tell, a trick to share, a technique to teach, we want you to consider joining the family of those who write for *MacTech*.

Don't just wait for a topic to be covered or a technique explained in *MacTech!* Take responsibility! Write us an article yourself!

To write for *MacTech*, just send for our Writer's Kit. It's a Microsoft Word file containing the Styles you need to use, and giving lots of helpful advice and information, including all the legal stuff. You can let us know what you're writing about, or, if you want to, you can just write the article and spring it on us when it's done. [**Note:** We also have a need for people willing to make themselves available to write occasional product/book reviews.] If we publish your article, you'll be paid for it!

Write to me, Matt Neuburg, at editor@mactech.com (or one of the other editorial addresses listed on page 2 of the magazine). Take the future of *MacTech Magazine* into your own hands!

For Macintosh Programmers & Developers
MacTech™ MAGAZINE

# Make more money than your parents ever thought you could.

Programmers Wanted
MACXPERTS
software design & engineering
"When I'm not writing C++ they send me to the Bahamas."
Programmers and Adventurers Wanted
C, C++, MacApp, OpenDoc, Java, NTK, 4D with basic to advanced knowledge of the Macintosh Toolbox.
Great environment, salary, benefits, 401K.
Call Mike about MT8 recruiting ad.
1-800-356-8040 fax / 804-358-3847
http://www.macxperts.com/~xperts
Email:xperts@infi.net or AOL:MacXperts

**Call MacXperts today.**

MacXperts provides its clients with the highest quality, best designed Macintosh, Newton, OpenDoc, Internet, and Intranet custom software.

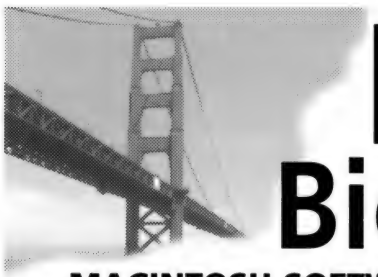
And we provide our programmers with the best, too. We offer excellent salaries, a 401K retirement plan, full medical, full dental, disability insurance, profit-sharing, stock appreciation, two weeks vacation, ten paid holidays (including your birthday), flex-time, all the latest development tools, and a great environment to pursue your interests.

**Call MacXperts 1-800-356-8040 fax / 804-358-3847**

**http://www.macxperts.com/~xperts email: macxperts@aol.com**

MACXPERTS
*software design & engineering*

---



WHAT YOU SHOULD KNOW ABOUT

# Bio-Rad

## MACINTOSH SOFTWARE DEVELOPER

*For nearly 40 years, Bio-Rad Laboratories has been recognized as a leading multinational developer, manufacturer and distributor of life science research products, clinical diagnostics and analytical instruments. With annual sales approaching $400 million, Bio-Rad's impact extends beyond research and industry, ultimately enhancing the quality of life around the world. We credit our unparalleled success to the efforts of our staff—dedicated, talented people from diverse backgrounds and specialties.*

Opportunities are available in our Northern California R&D facility (located just North of San Francisco) for a Macintosh Software Developer to join a small team of engineers, scientists, marketers and sales representatives in creating state-of-the-art image analysis tools for genetic research.

This position requires skills in Macintosh application development, and C/C++ programming. MacApp 3.x and OMT experience a plus. BSCS/EE, or equivalent and 4+ years' experience in systems and/or applications software development also required.

*Bio-Rad offers a compensation and benefits package you would expect from an industry leader. To be considered, please mail, fax or e-mail your resume, indicating Job Code, to: BIO-RAD LABORATORIES, 2000 Alfred Nobel Drive, Hercules, CA 94547. FAX: (510) 741-5803. E-mail: lsg_employment@bio-rad.com. An EEO/AA Employer. Principals only, please.*

BIO-RAD

---

## Software Developers – Cornell University

Join the team that brought you CU-SeeMe! The Advanced Technology group in Cornell University's campus networking organization is looking for talented programmers who want to be part of forward-looking, Internet oriented, and history making projects.

Cornell, a leading research university, is located in Ithaca, a small city in the Finger Lakes district of upstate New York, free from urban strife and blight and close to a wide range of recreational opportunities. But as home to both Cornell and Ithaca College, Ithaca is unusually cosmopolitan and active, intellectualy and culturally, for a community that provides such easy access to a rural or small town lifestyle.

Minimum requirements include a Bachelor's degree and two-three years of experience (or equivalent combination). Needed are skills in C, C++ (and Java), realtime, network oriented systems and applications development.

Experience needed in several of: graphics, video, audio, driver, network, and GUI development on Mac, Windows, NT, and Unix. Preference given to highly productive, creative, quick learners with teamwork experience. Generous benefit package. Several openings. Position level and compensation depending on experience.

Cornell University is an equal opportunity, affirmative action educator and employer.

Interested candidates should contact:
Richard Cogger
Cornell Information Technologies
110 Maple Avenue, Ithaca, NY 14853
or by e-mail rc19@cornell.edu

# Point your world of Macintosh experience in our direction.

Microsoft is committed to providing the best products possible for those who make Macintosh® their platform of choice. Our team is developing next-generation applications and technologies which will harness the excitement and the power of the Internet for Mac users everywhere. If you have demonstrated experience developing, testing or marketing Mac products, think about what you could achieve with the resources of a company like Microsoft behind you.

### Software Test Lead, Web Authoring

Design and develop test scripts for our new web authoring product, FrontPage™; supervise a small automation team and assist team members with automation development; and maintain the performance and client/server testing lab. Requires 5+ years software testing experience on commercially shipped Macintosh or Windows® products; knowledge of C/C++; and previous GUI automation experience. QA Partner, Microsoft Test and client/server experience preferred, as is knowledge of Microsoft Office products.

### Program Manager, Web Authoring

Drive development of FrontPage, from design and implementation through release. Involves writing functional specifications; defining how engineering teams are to implement functionality; overseeing products through development, testing and localization; and owning the schedule to ensure the product ships on time. Requires 3+ years software development experience, ideally in a project management capacity, and demonstrated ability to design easy-to-use interfaces. Knowledge of HTML, web editing, wizards and Microsoft Office applications preferred.

### Software Design Engineer

Develop Internet products such as browsers, news/mail/IRC clients, script/component and Java vm/runtime. Design dynamic media products that leverage Windows formats on the Macintosh platform, and help create an Internet library providing services to both internal Microsoft groups and third party ISVs. Requires demonstrated Macintosh or Windows development experience, with specific knowledge in one of these areas: Internet clients/standards, graphics, text editing, networking (TCP/IP), scripting, components, QuickTime, animation, sound or user interfaces.

### Product Manager, Macintosh Office

Develop and manage the marketing and business strategy for Microsoft's Macintosh Office applications. Involves developing retail, direct marketing, reseller and OEM channels for this suite of applications. Requires 3+ years technical marketing experience, which includes marketing end-user applications for the Macintosh. A dedicated customer orientation and strong public relations skills essential.

▼

Microsoft offers competitive salaries and excellent benefits. Opportunities are available in both Seattle and the Silicon Valley. Please email your resume in ASCII text format to **resume@microsoft.com** (Indicate Dept. Am3c8-0896 in the subject header). Or mail it to: **MICROSOFT CORPORATION, Attn: Recruiting, Dept. Am3c8-0896, One Microsoft Way, STE 303, Redmond, WA 98052-8303.** No phone calls please. We are an equal opportunity employer and support workforce diversity.

**Microsoft®**

WHERE DO YOU WANT TO GO TODAY?™

**http://www.microsoft.com/jobs/**

## Be sure to look for Microsoft at MacWorld!

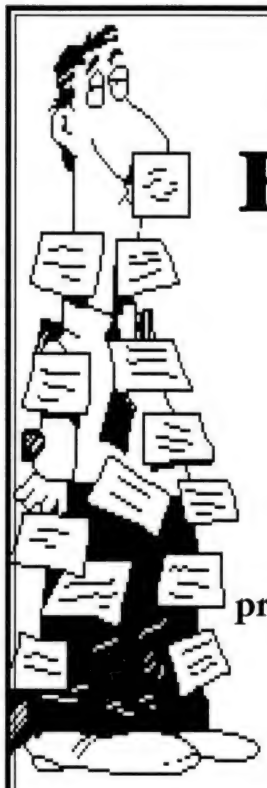
Microsoft and Windows are registered trademarks and FrontPage and *Where Do You Want To Go Today?* are trademarks of Microsoft Corporation. Macintosh is a trademark of Apple Computer, Inc.

*Continued from page 83*

can be separately upgraded without forcing a recompile of the project. The developer could even contract some components out, while keeping the core object and components in-house. The flexibility of components allows some powerful marketing opportunities, and the architecture of OpenDoc and SOM supports much quicker time-to-market with custom solutions.

Apple, IBM and the other OpenDoc partners have delivered a powerful architecture for developers to finally realize the promise of object technology – reuse, plug-and-play integration, and smooth evolution. Now we need tool developers to seize the day and make this architecture as accessible as the stand-alone application. Then perhaps, some day soon, reusable, high-quality educational components will be as familiar to every child as their first Lego kit, model train set, or hi-fi stereo system.

**Visit MacTech Magazine's Web site!**
http://www.mactech.com



**DON'T FORGET!**

**To receive information on any products advertised in this issue, send your request via Internet: to productinfo@xplain.com**

TestTrack

**Got a Minute?**

**Because that's all you need to start tracking bugs with TestTrack.**

**1-2-3 Start Tracking** TestTrack is ready to use right out of the box. Simply install, add a few users, and start tracking. It's that easy.

But TestTrack is more than just an easy-to-use bug tracking program—it's a powerful quality control tool for busy software development teams:

**Automate** TestTrack automates the tedious and error-prone process of reporting and tracking bugs by hand. It also eliminates the effort needed to create a custom solution using general purpose database tools such as 4D™ or FileMaker Pro™.

**Communicate** TestTrack links engineers, testers, managers, even tech writers together so no one falls out of the loop. Team members are notified automatically when defects are assigned to them, guaranteeing communication and ensuring efficient work flow.

**Stay up to Date** TestTrack lets any authorized user look up the current state of any defect at any time.

**Analyze** TestTrack makes reporting easy—point, click, print and read. Customize reports to list only what you want to see.

**To order** call 513-683-6456

**For more information**
sales@seapine.com
http://www.seapine.com

**For a limited time, you can buy TestTrack for the introductory price of $99!**
**Subscriptions, volume pricing, and site licenses are also available.**

Seapine Software

Seapine Software, Inc. 1066 Seapine Ct. Maineville, OH 45039 ◆ 513.683.6456 ◆ sales@seapine.com ◆ http://www.seapine.com

*By John Kawakami*

### Free Education on the Web

In ancient Greece, young aspiring philosophers would organize themselves into gangs, find the local know-it-all, and offer gifts and money in exchange for edification. This tradition of unaccredited, self-organized education continues on the computer networks, where newbie programmers clamor for advice from wiser, more experienced developers. Lately, however, online education has started to organize itself more formally.

Here and there on the Web and on the online services, established educational institutions have implemented accredited "distance learning" projects, with varying degrees of success. At the other end of the educational spectrum, numerous independent writers and editors have created the "canon" of online self-education: online tutorials, online books, and numerous FAQs. Somewhere between these two extremes, a handful of active users have organized interested people to seek out an education online, or get assistance to educate themselves.

Last year, Stephen Jovanovic set up the PowerPlant Dream Team, organizing dozens of programmers into groups to study and then develop in the PowerPlant framework. This project has grown into a new project: the CodeWarrior User Group, which has been officially recognized by Metrowerks. Part of the group's mission is to organize "Dream Teams" to study Java, C++, ODF/OpenDoc and PowerPlant. The aim of the Dream Teams is to foster learning in a collaborative, fun environment, and to help small groups of programmers produce new applications.

The CodeWarrior User Group is seeking volunteers and sponsors. For more information, contact Steve Jovanovic at stevej@lakesoft.com.

http://petrified.cic.net/powerplant-dt/

radar pangean [*yes, that's his name – jk*] has been developing a Web-based service called "The MOST", where mentors will help neophytes learn aspects of Macintosh programming. The MOST is more like a school than the Dream Team, because there is a prescribed course of study, and each group has a leader who acts as a mentor.

As their Web page describes it, the MOST Web site and services are the gifts of altruistic Macintosh professionals and amateurs to anyone – high school or college student, other platform developer, current Mac OS developer – who wants convenient access to resources that will help her produce high quality Macintosh applications.

The MOST provides an umbrella non-organization which facilitates opportunities for its volunteer members to work together to provide free access to mentors, information resources, self-study courses, and net-based, mentored classes. The courses and classes which The MOST produces will cover as many aspects of programming for the Mac OS as can be created (and then supported) in the volunteers' "spare time". The MOST makes its information and services available at no charge.

The MOST is seeking volunteers and sponsors. For more information, contact radar pangean at radar@SMT-Inc.com.

http://www.aimed.org/most/

The online services have also run ongoing classes teaching programming. America Online has an extensive archive of C programming course material. CompuServe is running classes on, among other things, Java. For more information about these programs, contact your online service provider.

### Conceptual Design Releases EightyRez Editor for AppleScript Dictionaries

Gary McGath, dba Conceptual Design, is releasing a new Macintosh application, EightyRez, for developers of scriptable software. This application allows easy editing of `'aete'` and `'aeut'` resources, also known as AppleScript dictionaries.

Until now, there have been no full-featured tools for editing AppleScript dictionaries. Various freeware tools have been developed, but all have drawbacks. EightyRez is a supported, commercial product to edit dictionary resources.

Features of EightyRez include:

- Viewing resources as collapsible outlines
- Stationery files for common suites
- Cutting and pasting whole suites or individual elements
- Exporting dictionary listings and Rez source files
- Pop-up menus for selecting commonly used values
- Printed documentation and Apple Guide help
- E-mail support

EightyRez requires Mac OS 7.0 or higher and a 68020 or higher processor. Eight megabytes of memory are recommended.

The purchase price is US$40, plus US$2 shipping within the U.S. and Canada, or US$10 shipping elsewhere or for second-day air delivery in the U.S. EightyRez is not for sale to the U.S. government or its agencies, in view of its ongoing hostility to our liberties, particularly the Communications Decency Act and restrictions on cryptographic software [*their comments, not mine – jk*]. Site licenses are negotiable. A demo is available at the Web site below.

http://www.mv.com/biz/condes/EightyRez.html

### MaxumPack

Maxum Development Corp. has released the MaxumPack 1.0, an application bundle to make your Macintosh Web site dynamic, interactive, reliable, secure and fast. Included in the bundle are current versions of all Maxum's shipping products: NetCloak, NetForms, PageSentry, WebLock and RushHour.

NetCloak 2.0 extends the capabilities of a Macintosh-based Web server by adding a total of 45 commands for use in World Wide Web pages. It executes these commands "on-the-fly" as the Web pages are sent to the client. Web authors can customize their pages by the following values: client domain, time, date, browser, referrer, username, password, access count, any client-entered information, and a random number. NetCloak dynamically serves graphics, text, hyperlinks and other HTML elements and includes advanced features like macros and high-performance caching.

NetForms is a Web server "back-end" that allows Webmasters to easily process information contributed by visitors to create completely interactive Web sites. NetForms can take user-submitted information and create a new HTML page, send an e-mail message, insert text into an existing page, and store the data for later processing by virtually any application (including databases, spreadsheets, text-processors, etc.). The Webmaster is in control of how the page is formatted, how the menus are kept, and what words or phrases will be linked automatically.

PageSentry 1.1 will monitor any page, on any server, anywhere on the Internet. PageSentry allows the Webmaster to create "sentries" that perform complete HTTP file transfers. The files are checked for valid content and, in the case of a failure, the Webmaster is notified. By performing real HTTP transfers, PageSentry verifies that all parts of the Web server are functioning, including the operating system, the TCP/IP stack, the Web server itself, and even CGI applications.

WebLock is a performance and security utility incorporating RAM-based file caching and security based on the Mac OS built-in File Sharing and Users and Groups. Unlike security schemes based on realms, which require complex naming conventions and can have undesirable side-effects, WebLock allows you to protect folders with a simple click.

RushHour is a software-based Web server graphics accelerator for the Macintosh which eliminates performance bottlenecks that typically are responsible for significant server slow-downs. RushHour runs in conjunction with your existing Web server software and allows you to improve dramatically the overall throughput at your site, even on extremely busy sites that serve large files. As graphics comprise the majority of most Web server traffic, RushHour may return a "three- to ten-fold" improvement in performance.

The MaxumPack carries a list price of US$695 with a special introductory price of US$595 through July 31, 1996. Maxum Development. Tel: (708) 830-1113.

http://www.maxum.com/

### FairCom

FairCom Corporation announced the release of its new Developer CD, which features the new c-tree Plus version 6.6A and FairCom Macintosh Server release 6.06.26B. The FairCom Developer CD also gives the developer fully functional FairCom database servers for over 25 supported environments. In addition, the CD offers full on-line documentation with complete search capabilities.

"FairCom supplies efficient and stable code that I can count on when I need to port it to a new compiler variant," said H. Tom Thomas, President of Tetra Systems, Santa Fe, New Mexico, which provides conference abstract management for companies like Monsanto. "With compilers getting updated every four months, it's nice to know making c-tree work won't be a problem. For example, porting c-tree to the latest Microsoft cross-compiler for Macintosh was simple," Thomas explained.

The CD also supports development on Windows 3.1, Windows 95, and Windows NT with new automatic windows instance logic, native NT support for the Dec Alpha and NEC MIPS CPUs, and a native Windows 95 Server.

Other c-tree Plus File Handler features include a low level file flush function for flushing file caches, a special configuration function providing pertinent information about the c-tree Plus application, new memory allocation tracking abilities, and many other functional enhancements. FairCom Server enhancements include executing user applications during Server startup and shutdown, a complete system administration API which allows developers to perform system administrator duties from within C programs, additional control and flexibility over the Server automatic recovery process, server speed enhancements due to reductions in network traffic, and additional memory control options.

The FairCom Developer CD with c-tree Plus V6.6A is priced at US$895 with full C source code, no royalties, 25+ free development Servers. Licenses are sold per-developer. FairCom. U.S. tel: (800) 234-8180; Europe tel: (039) 35-773-464; Japan tel: (0592) 29-7504.

http://www.faircom.com/

### Important Reminder

Information for Newsbits should be submitted via email, and sent to press_releases@mactech.com (not to any of the other MacTech email addresses, please!).

Also, don't forget our moderated news mailing list, MacDev-1™, which is distributed to thousands. To a submit a posting to MacDev-1, just send it as email to MacDev-1@listmail.xplain.com. (To become a recipient of MacDev-1 mailings, send mail to macjordomo@listmail.xplain.com, with "`sub MacDev-1 <yourname>`" in the body of the message.)

# TIPS & TIDBITS

*By Steve Sisak*

## TIP OF THE MONTH

### CodeWarrior Target Switch

Here's an easy way (in CodeWarrior) to build a project file for another target if you have been working on one target already (i.e., working on a 68K project, and now it's time to make a PPC project).

Even though I create apps for both 68K and PPC, I always just work in a 68K project file, since I have a 68K Mac at home to work on.

When it comes time to make a PPC binary for testing, or just to see if the thing will compile with a PPC compiler, I'll go to the Finder and duplicate (**⌘D**) my 68K project file, and then rename the copy to something like "MyProject (PPC)".

Open the new project up, reset all the project prefs to a PPC target, set other prefs as needed, remove the 68K libraries, and add in the PPC libraries (and you might need to update the precompiled header files, unless you use something like a **Prefix.h** file that conditionally adds in the right precompiled header based upon the target).

Violà! It's a pretty simple thing that saves you lots of time in recreating project files for multiple targets.

*John Daub*

### The Origin of the GWorld

Sometimes, images drawn into offscreen GWorlds for a translucent drag don't draw properly if the GWorld's portRect starts at any location other than (0,0). This can result in items being clipped improperly, or not being drawn at all.

This problem only occurs when the user has multiple monitors hooked up.

Translucent drags will work fine if you create the GWorld for the drag image at location (0, 0).

*Jeremy Vineyard*

### But How About Free Overnight Shipping?

If you'd like to learn more tricks on optimizing PowerPC assembly language code, call IBM and order the 240-page *PowerPC Compiler Writer's Guide*. They'll charge you $1 (yes, one dollar) and you have to pay with a credit card (at this price you should order four or five for friends and family). The phone number is (800) 879-2755 (IBM's Manuals & Publications). The book is ISBN 0-9649654-0-2 and IBM part number MPRPPCOMP-01.

Here's an example from their Optimal Code Sequences chapter. The C code is calculating the absolute value of v0:

```
absoluteValue = (signed_word) v0 < 0 ? -v0 : v0;
```

The optimal instructions (which are nice to the pipeline, no branches) are:

```
srawi     R4, R3, 31  // v0 is in R3 on entry
add       R5, R4, R3
xor       R6, R5, R4
```

There are about 40 other short sequences like this one given in the book, as well as discussions on instruction scheduling, alignment, etc. Well worth the price!

*Mike Scanlin*

*Send us your tips or we'll install EvenBetterBusError on your machine! On the other hand, we might just pay you **$25** for each tip we use, or **$50** for Tip of the Month. You can take your award in goods, subscriptions or US$. Make sure any code compiles, and send tips (and where to mail your winnings) to our **new Tips e-mail address** at tips@mactech.com. (See page 2 for our other addresses.)*

With MacTech™ Magazine articles in THINK Reference™ format.
Includes THINK Reference 2.0

# Bigger, Better, Faster!

Introducing the new MacTech CD-ROM™ Vol. 1-11. Get all 127 issues of *MacTech Magazine*™ plus a new THINK Reference™ for faster access, all the source code, working applications, demos, frameworks, and more – all in one CD!

- 1420+ articles, from all 127 issues of *MacTech Magazine* (1984 – 1995).
- Improved hypertext, and a new THINK Reference Viewer – for lightning-quick access.
- New super-fast word search.
- 100+ MB of source code – use them in your own applications, with no royalties.
- Full version of THINK Reference 2.0. The original online guide to *Inside Macintosh*, Vols. I-VI.
- 80MB of FrameWorks/SFA archives. The most complete set of FrameWorks archives known.
- Sprocket™! MacTech's Tiny Framework that compiles quickly and supports System 7.5 features.
- The best threads from the Macintosh programmer newsgroups, plus thousands of notes, tips, snippets, and gotchas.
- Popular tools that Macintosh programmers use to increase their productivity.

Available Through Developer DEPOT™

## Now Only $49!

List Price: $89

$39 for upgrades from previous versions.

(prices do not include shipping and handling)

http://www.devdepot.com • Phone: 800-MACDEV-1 • 805-494-9797 (outside the U.S. and Canada)
Fax: 805-494-9798 • E-mail: orders@devdepot.com

# ADVERTISER/ PRODUCT LIST

## LIST OF ADVERTISERS


**A. D. Software** ..... 34
**Absoft** ..... 66
**Addison-Wesley Publishing Co.** ..... 21
**Adianta Inc.** ..... 44
**ADInstruments** ..... 80
**Adobe Developers Association** ..... 83
**Aladdin Knowledge Systems Ltd.** ..... 5
**Aladdin Systems, Inc.** ..... 12-13
**AP Professional** ..... 68
**Apple Developer Relations** ..... 24-25
**Apple Developer University** ..... 85
**Bare Bones Software, Inc.** ..... 37, 48
**Bio-Rad Laboratories** ..... 87
**Bowers Development** ..... 11
**BroadCast Software Corp.** ..... 62
**Celestin Company, Inc.** ..... 82
**Conceptual Designs** ..... 70
**Cornell University** ..... 87
**Cyclos** ..... 63
**Datapak Software Inc.** ..... 76
**DC Micro Development** ..... 83
**Developer Central Macworld** ..... 61
**Digitool** ..... 85
**dtF Americas, Inc.** ..... 35
**EveryWare Development Corporation** ..... 52
**Faircom Corporation** ..... 57
**FGM, Inc.** ..... 44
**Jasik Designs** ..... 45
**JointSolutions Marketing** ..... 72
**JYACC** ..... IBC
**Kaidan** ..... 74
**MacTech CD-ROM 1-11** ..... 93
**MacTech Web™ Site** ..... 69
**MacXperts** ..... 87
**Main Event** ..... 39
**Mango Tree Software, Inc.** ..... 47
**mark/space Softworks** ..... 78
**Mathemaesthetics, Inc.** ..... 15
**Metrowerks** ..... BC
**Micro Macro** ..... 9
**Microsoft** ..... 7, 28, 88
**MindVision Software** ..... 41, 43
**Miracle Software, Inc.** ..... 71
**Motorola** ..... 27
**Natural Intelligence** ..... 64
**Neologic Systems** ..... 17
**Nisus Software, Inc.** ..... 47
**Now Software** ..... 81, 82
**Onyx Technology** ..... 80
**OPtions Computer Consulting** ..... 74
**PACE Anti-Piracy** ..... 72
**Quarterdeck Corporation** ..... 58, 59
**Quasar Knowledge Systems** ..... 1
**Ray Sauers Associates** ..... 77
**Release Software Corporation** ..... 23
**Scientific Placement** ..... 86
**Seapine Software, Inc.** ..... 89
**StepUp Software** ..... 67
**Stone Tablet Publishing** ..... 43
**Supersoft** ..... 79
**Symantec** ..... IFC
**Terran Interactive** ..... 32
**The Trattner Network** ..... 87
**Uni Software Plus GMBH** ..... 10
**Water's Edge Software** ..... 84
**Willows Software Inc.** ..... 19


## LIST OF PRODUCTS


**A/Pay** • Release Software Corporation ..... 23
**Adobe Developer Tools** • Adobe Developers Association ..... 83
**Apple World Wide Developer Web Site** • Apple Developer Relations ..... 24-25
**AppMaker** • Bowers Development ..... 11
**Apprentice 4** • Celestin Company, Inc. ..... 82
**BBEdit** • Bare Bones Software, Inc. ..... 37
**BroadCast™** • BroadCast Software Corp. ..... 62
**C/C++ SDK** • Motorola ..... 27
**CodeWarrior™** • Metrowerks ..... BC
**Crusher!™ Data Compression Tool Kits** • DC Micro Development ..... 83
**c-tree Plus®** • Faircom Corporation ..... 57
**The Debugger V2** • Jasik Designs ..... 45
**DragInstall 2.0** • Ray Sauers Associates ..... 77
**dtF, The Relational Database System** • dtF Americas, Inc. ..... 35
**EightyRez** • Conceptual Designs ..... 70
**Fortran SDK** • Motorola ..... 27
**FairCom® Developer CD** • Faircom Corporation ..... 57
**Fortran 77 for Macintosh** • Absoft ..... 66
**4D Tool Kit** • OPtions Computer Consulting ..... 74
**Guide Composer** • StepUp Software ..... 67
**Installer VISE 4.0** • MindVision Software ..... 41
**ListSTAR** • Quarterdeck Corporation ..... 52
**JAM®** • JYACC ..... IBC
**MacApp Programmers** • MacXperts ..... 86
**MacHASP** • Aladdin Knowledge Systems Ltd. ..... 5
**MacNosy** • Jasik Designs ..... 45
**MacRegistry™** • Scientific Placement ..... 87
**MCL** • Digitool ..... 85
**The Memory Mine™** • Adianta Inc. ..... 44
**MicroGuard Plus™** • Micro Macro ..... 9
**Movie Cleaner Pro™** • Terran Interactive ..... 32
**neoAccess™** • Neologic Systems ..... 17
**neoShare™** • Neologic Systems ..... 17
**Now Up-To-Date** • Now Software ..... 81
**Now Utilities™** • Now Software ..... 82
**OOFile™** • A.D. Software ..... 34
**OpenDialog™** • FGM, Inc. ..... 44
**PACE Software authorization systems** • PACE Anti-Piracy ..... 72
**PAIGE** • DataPak Software, Inc. ..... 76
**Publications** • Addison-Wesley Publishing Company ..... 21
**QC** • Onyx Technology ..... 80
**QUED/M™ 3.0** • Nisus Software, Inc. ..... 47
**Quicktime VR Quicktime** • Kaidan ..... 74
**Recruitment** • Bio-Rad Laboratories ..... 87
**Recruitment** • Cornell Univeristy ..... 87
**Recruitment** • MacXperts ..... 87
**Recruitment** • Microsoft ..... 28, 88
**Recruitment** • Scientific Placement ..... 86
**Recruitment** • The Trattner Network ..... 86
**Resorcerer® 1.2** • Mathemaesthetics, Inc. ..... 15
**Roaster™** • Natural Intelligence ..... 64
**Scripter® version 2.0** • Main Event ..... 39
**Smaller Installer** • Cyclos ..... 63
**SmalltalkAgents** • Quasar Knowledge Systems ..... 1
**SoftPolish** • Bare Bones Software, Inc. ..... 48
**StuffIt InstallerMaker 3.0** • Aladdin Systems, Inc. ..... 12-13
**Stone Table** • Stone Tablet Publishing ..... 43
**SuperPlot™** • SuperSoft ..... 79
**Symantec C++** • Symantec ..... IFC
**Tango** • EveryWare Development Corporation ..... 53
**TCP/IP Scripting Addition** • Mango Tree Software, Inc. ..... 47
**TestTrack™** • Seapine Software, Inc. ..... 89
**TMon Professional** • Mindvision Software ..... 43
**Tools Plus™ 3.0** • Water's Edge Software ..... 84
**Twin APIN** • Willows Software Inc. ..... 19
**UpdateMaker 2™** • ADInstruments ..... 80
**Updater VISE 1.1** • MindVision Software ..... 41
**Visual FoxPro** • Microsoft ..... 7
**Voodoo** • Uni Software Plus GMBH ..... 10
**WebWeaver** • Miracle Software, Inc. ..... 71
**World Wide Web Pages Creative** • JointSolutions Marketing ..... 72
**WebSTAR** • Quarterdeck Corporation ..... 52



The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

Mac OS
Scripter
Apple Guide Starter Kit
Danny Goodman
Jeremy Joan Hewes
FRAGMENT OF YOUR IMAGINATION
Code Fragments Resources for PowerPC and Macintosh
Joe
SIC for the Newton
John Schettino Liz O'Hara
New Technology Building Blocks
Macintosh Programming Techniques
A Foundation for All Macintosh Programmers
Learn essential techniques for expert Macintosh programming
Understand terms and concepts unique to Macintosh programming
Gain hands-on experience with interactive software tutorial
Dan Parks Sydow
LEARN C ON THE MACINTOSH
SECOND EDITION
SUBSCRIPTION VERSION
SYMANTEC.
C++
DEVELOPMENT SYSTEM FOR POWER MACINTOSH
CodeWarrior
FACESPAN
The Interface Designer and Application Builder.
Dave Mark Cartwright Reed
Macintosh C Programming PRIMER
Volume I
Second Edition
Inside the Toolbox Using THINK C
OPTIMIZ
Power
CODE
Gary Kacmar
leScript
Additions Guide
Essential
Developer
DEPOT

# Going to MacWorld?

## Don't forget to bring a toothbrush, clean socks, your PJs... *and a shopping cart!*

*Oh yeah!* Because this year, we're inaugurating the MacWorld Developer Depot™ booth, and we're kicking it off *with a bang!* Tons of stuff is going on sale, and if you think our prices are low now, wait 'till you see what we've got in store for you.

With insane discounts of up to 45%, and humongous package deals, you'll want to restock your entire arsenal. We'll be offering all the usual great stuff, plus an onslaught of awesome new products – *all at the lowest prices guaranteed!*

And as always, we'll be the exclusive distributors of all MacTech™ products.

So if you're planning a trip to Boston, come prepared. And if you can't get a hold of a shopping cart, we'll be happy to provide you with our sturdy Developer Depot bags.

See you there!

**Developer DEPOT™**

**http://www.devdepot.com**

800/MACDEV-1 • Outside the U.S. & Canada: 805/494-9797 • Fax: 805/494-9798
E-mail: orders@devdepot.com • Web Site: http://devdepot.com

Dear Developer,

Since March, 1985, **MacTech**™ Magazine (then MacTutor) has been the place for you to find Macintosh Development products – the majors as well as those harder to find. Back then … and today, our focus has always been for the Macintosh developer community to grow and prosper.

But you deserve more than simple listings – and you've asked to make it easier to find and order products while maintaining our renowned customer service and pricing. We've been listening …

We are therefore proud to announce **Developer Depot**™ – a new mail order concept and entity dedicated to you, the Mac OS Developer. What makes Developer Depot so good?

- World renowned customer service
- 24-hour and full accessibility via our continually updated web site (http://www.devdepot.com)
- Satisfaction and best price guaranteed
- A new, easy to remember toll free phone number (**800-MACDEV-1**)
- Great selection (hundreds of products)
- Convenience: you can order in many ways (web site, e-mail, fax, phone)
- A new full color, organized, and expanded catalog – updated monthly

Additionally, we are a Macintosh house using an all Macintosh system. Our knowledgeable staff can answer your questions about products – and if they can't, they'll find the answers for you.

While Developer Depot is completely separate from MacTech Magazine, Developer Depot is the primary source for all MacTech products.

Let us know what you think – we're here to serve you.
**Developer Depot**: The products you need with the service and prices you deserve.

Neil Ticktin
Chief Executive Officer

Andrea J. Sniderman
Chief Operating Officer

# TABLE OF CONTENTS


02 MACTECH
03 DEVELOPMENT ENVIRONMENTS
06 INTERNET RELATED
07 SCRIPTING
08 TOOLS, LIBS & UTILITIES
14 BOOKS & REFERENCE

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

**Developer Depot 30 day Money Back, Price and Satisfaction Guarantee**
Developer Depot products are sold with a 30 Day money back guarantee on user satisfaction, lower prices and against defects. If, for any reason, you are not satisfied or find the same product at a lower price within 30 days, please call Customer Service at 800-MACDEV-1 and request a Return Merchandise Authorization (RMA) number to get a full refund or the difference in price (where applicable). You must return undamaged product at your expense, including all its original packaging, documentation and the blank warranty card if applicable. Developer Depot will replace defective product upon receipt of the defective merchandise. Please remember to back up your data before installation of any new hardware, software, or peripherals; we cannot be responsible for any lost data. Policies, item availability, and prices are subject to change without notice. The price in effect when we receive your order will be the price that you are charged. We are not responsible for any typographical errors in this or any other catalog, nor for any misstatements from any vendor. Purchase orders are also accepted but must be in writing, signed, come with a contact person and a telephone number, and mailed to P.O. Box 5200, Westlake Village, CA 91359-5200. Faxed copies and purchase order numbers alone are not acceptable.


Stuff our lawyer made us write.
Developer Depot makes no other warranties. All other warranties, either expressed or implied, including the implied warranty of merchantability and fitness for a particular purpose are disclaimed. Developer Depot shall not be liable for any direct, special, incidental or consequential damages including lost profits, from any delay in delivery, or for any personal injury arising from the use of any product sold through Developer Depot. The limit of direct damages, if any, shall not exceed the purchase price of the product.© 1996 Xplain Corporation. All rights reserved. Any unauthorized duplication is in violation of federal laws. Developer Depot is a registered trademark of Xplain Corporation. All product names in this catalog are the trademarks of their respective holders.

© 1996 Xplain Corporation. All rights reserved. Any unauthorized duplication is in violation of federal laws. Developer Depot is a trademark of Xplain Corporation. All product names in this catalogue are trademarks or registered trademarks of their respective holders

For Macintosh Programmers & Developers

MacTech™ MAGAZINE

## MacTech Magazine

Subscriptions: US magazine:
**$47** for 12 issues (MTYRDM)
Canadian: **$59** for 12 issues (MTYRCM)
International: **$97** for 12 issues (MTYRFM)
**Back Issues**: **$5** each (subject to availability)

## MacTech CD-ROM Volumes 1-11 NEW!

- 1420+ articles, from all 127 issues of MacTech Magazine (1984-1995).
- Improved hypertext, and a new THINK Reference Viewer — for lightning quick access!
- New hyperlinks between articles allow you to jump to other relevant articles.
- 100+ MB of source code — use them in your own applications, with no royalties!
- Full version of THINK Reference™ 2.0. The original online guide to Inside Macintosh, Vols. I-VI.
- 80MB of FrameWorks/SFA archives. The most complete set of FrameWorks archives known.
- Sprocket™! MacTech's Tiny Framework that compiles quickly and supports System 7.5 features.
- The best threads from the Macintosh programmer newsgroups plus thousands of notes, tips, snippets, and gotchas.
- Popular tools that Macintosh programmers use to increase their productivity and much more!

Our Price **$89.00** (SMTCD11) Upgrades: Our Price **$39.00** (SMTCD11U)

NEW!

## MacTech Mouse Pad

Slide on this! With an extra-large surface (11" by 10") and a deluxe sleek plastic coating, you'll be zooming across your screen in no time at all.
Speed limit not enforced!

Our Price **$8.95** (AMTPAD)

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

## EXCLUSIVES!

### FrameWorks Magazine

$8 per back-issue, subject to availability. (MTFWBACK)

### FrameWorks Source Code Disk

$10 per back issue, subject to availability. (MTFWSC)

### MADACON '93 CD-ROM

The highlights of MADACON '93, including Mike Potel on Pink, Bedrock, MacApp, OODLs, and more. Slides, articles, demos, audio, and QuickTime.

List $95.00 Our Price **$9.95** per set (SMADA93).

BLOW OUT!

### Best of MacTutor

The Best of MacTutor Collection, Vols.3-5.
List $99.00 Our Price **$9.95** per set (BMTBEST).

## CodeWarrior 9 Gold by Metrowerks

- Build applications for the Mac OS, Windows 95, Windows NT, Magic Cap, & the new Be OS.
- Includes C, C++ and Object Pascal compilers, source-level debuggers, object-oriented frameworks (PowerPlant, MacApp, MFC), Apple's MPW, complete online documentation and source code examples for all languages and platforms.
- The IDE software has been localized in eight languages plus English. CodeWarrior is sold on an annual subscription basis and has three major releases each year.

Our Price **$399.00** (SCWGOLD)

SEE RELATED CATEGORY: **Internet Related**

SEE RELATED PRODUCTS:
**• Code Manager • Inside Power Plant • Inside CodeWarrior 9 • C++ Programming with CodeWarrior • PowerMac Programming Starter Kit • Discover Programming for Macintosh • Learn C on the Macintosh • Metrowerks CodeWarrior Programming**

## CodeWarrior 8 Bronze

- Build applications for the Mac OS on 68K machines.
- Includes C, C++ and Object Pascal compilers, source-level debuggers, object-oriented frameworks (PowerPlant, MacApp), Apple's MPW, complete onlinedocumentation and source code examples for all languages and platforms.
- The IDE software has been localized in eight languages plus English.
- Sold on an annual subscription basis and has three major releases each year.

Our Price **$149.00** (SCWBRON)

## CodeWarrior Wear (XL only)

**You live it, you breath it... you might as well wear it!**

- Black CodeWarrior Sweatshirt: Our Price **$29.95** (ACWSWEAT)
- "Blood, Sweat & Code" black long-sleeve shirt: Our Price **$14.95** (ACWBLOOD)
- Cross Platform white shirt (not pictured): Our Price **$14.95** (ACWXPS)
- Hawaii Five-0 shirt (not pictured): Our Price **$7.95** (ACWHAWAII)
- Arnold at work T-shirt (not pictured): Our Price **$9.95** (ACWARNLD)
- Hat (black or white): Our Price **$14.95** (ACWHAT)
- Winter Hat (not pictured): Our Price **$14.95** (AWINHAT)

## CodeWarrior Mouse Pads

Fight your way through the night with these cool-looking pads!

- Arnold: Our Price **$8.95** (ACWPADA)
- Gears (not pictured): Our Price **$9.95** (AMGEAR)

## Presenting Magic Cap, A Guide to General Magic's Revolutionary Communicator Software

- Perfect for novices as well as experts who want to take full advantage of Magic Cap.
- Step through its screens, see how it works, and learn ways to use it.
- Describes the power of smart messaging to customize the way you handle e- mail, a name card file that learns how to keep track of your contacts, a datebook that performs like a faithful assistant, and countless other abilities.

List $16.95 Our Price **$15.25** (BPRESMAGIC)

## SmalltalkAgents® by Quasar Knowledge Systems

• Features a new generation of the Smalltalk language, QKS Smalltalk™. • Productivity is no longer measured in lines of code, but in project completion time. • Allows "live" direct manipulation of your objects. • Provides a dynamic, interactive and iterative development process. • Provides easy and full access to the features of the Mac OS™ and Mac Toolbox. • Link your non-Smalltalk code resources using our External Code Linking Toolkit™ (ECLT). • A sophisticated database for source code management provides an almost infinite variety of ways to cross-reference, access, view, and manipulate your code and objects. • Includes an Application Delivery Toolkit™ (ADT) that allows you to create royalty-free, stand-alone, double-clickable applications in just a matter of minutes. • Any component built in AO/S will function as an OpenDoc component or container and components from non-AO/S sources can be seamlessly integrated into the AO/S system.

Our Price **$695.00** (SSTA)

SEE RELATED CATEGORY: **Internet Related**

SEE RELATED PRODUCTS: Symantec C++ Programming
SEE RELATED PRODUCTS: Learn C++ on the Macintosh
SEE RELATED PRODUCTS: Programming in Symantec C++
SEE RELATED PRODUCTS: Mastering theTHINK Class Library

## Symantec C++ 8.5

- Native for Power Macintosh.
- Develop full-featured Power Macintosh applications quickly and easily!
- Environment includes: a true native Power Mac implementation of the C++ language, MrC/MrC++ compilers for fast Power Mac executable code, Visual Architect for fast, easy GUI generation, a new THINK project management system that supports complex applications, a new editor/browser , a powerful, easy-to-use source code debugger.
- Also includes: The industry-standard THINK Class Library!
- Next two updates free when you send in your registration card and more!

List $499.00 Our Price **$349.00** (SSYMCPP)

## Symantec C++ for 68k

• The standard in development languages. • Powerful combination of fully integrated visual tools and the latest in C and C++ compiler technology. • Provides an object-oriented approach to application development. • Write code that is extensible, reliable, and maintainable. Includes: Integrated Environment with full source-code debugging, Incremental Linker which eliminates long link times, THINK Class Library, AppleEvents, Visual Architect™, THINK Inspector, Project Models, Source-Code Control with integration with Apple's SourceServer (included), Support for Scripting, Powerful Standard Libraries which includes I/O Streams, ANSI standard C library, and sample programs. • Full source code is included – Create applications that run on 68K Macs and Power Macs (emulated). These applications can easily be migrated to native Power Mac, by trading up to Symantec C++ for Power Mac.

List $299.00 Our Price **$99.00** (SSYMCPP68K)

## Think Pascal Version 4.0

**by Symantec Corporation**

- Great for professionals and students
- Fully integrated for rapid turnaround time – take advantage of System 7 capabilities
- Supports large projects, enhanced THINK Class Library, System 7 compatibility, superior code generation, and smart linking.
- Includes four Macintosh disks, a user manual, and an object-oriented programming manual.

Our Price **$169.00** (SPASCAL)

## LS Object Pascal CD-ROM by Fortner Research.

- Includes the world's first Object Pascal compiler for Power Macintosh
- 100% compatible with Apple's MPW Pascal
- Combines the best of Apple's native development tools with innovative new technology developed at Language Systems
- Compiler options specify 68K or native PowerPC code generation
- CD includes: LS Object Pascal compiler, Universal Pascal Toolbox interfaces, fully loaded MPW 3.3.1, 68K and PowerPC source debuggers, PowerPC assembler, online documentation, Macintosh Tech Notes, and a special version of AppMaker by Bowers Development that generates native Pascal source code.

List $399.00 Our Price **$349.95** (SOBJPASC)

## MachTen – Power UNIX by Tenon Intersystems

- Dynamically linked shared libraries, memory mapped file access and integrated UNIX and Macintosh development tools.
- BSD 4.4 and conforms to the Federal Information Processing Standard 151-2 (the POSIX FIPS).
- Pre-emptive multitasking for UNIX applications and includes a full featured high-performance TCP/IP protocol stack that supports multi-homing and multi-casting, features not yet available even with Apple's new Open Transport.
- A complete UNIX software development environment with a source-level debugger and C, C++, and Fortran compilers all generating native PPC code.
- Also included is a high-performance X server and complete X11R5 X.

Our Price **$695.00** (SM10PPC)

SEE RELATED CATEGORY: Internet Related

## Personal MachTen for 68K Macs

- MachTen UNIX for Macintosh (from Classic on up, including PowerBooks and Duos)
- A Mach-based Berkeley UNIX with pre-emptive multi-tasking
- Extends the Mac OS with UNIX networking and software development tools.
- Built-in internet services include domain name service, POP mail service, internet routing, SLIP & PPP, and Web service.

Our Price **$495.00** (SM10PER) Also Available: MachTen XWindows Our Price **$350.00**

SEE RELATED CATEGORY: Internet Related

## Mjølner BETA System

• Software development environment supporting object-oriented programming in the BETA programming language. • Uniquely expressive and orthogonal. • Unifies just about every abstraction mechanism – including class, procedure, function, coroutine, process and exception. • Includes: general block structure, strong typing, whole/part objects. The compiler: binary code generation, automatic garbage collection, separate compilation, interface to C, Pascal, and assembler. • The system: persistent objects, basic libraries with containers classes, platform-independent GUI application frameworks on Unix, Mac and Windows NT, metaprogramming system. • The Mjølner BETA System for Macintosh requires MPW (basic set) 3.2 or later. Package containing compiler, basic libraries, persistent store, GUI framework, and comprehensive documentation. (Other packages are also available).

List $295.00 Our Price **$245.00** (SMJOLNER)

## LS Fortran by Fortner Research

• 68K or PowerMac LS FORTRAN comes in two versions: one that generates 68K code (with or without FPU) and one that generates native code for the PowerMac • Source code is compatible between these two compilers • Programs originally developed with the 68K version can boost execution speed by 4 to 6 times by running the same program on a PowerMac • Supports all VAX intrinsic functions and data types, and virtually all VAX extensions • Built-in Debugging • Free Runtime Licenses • Full-featured editor, multiple window capabilities, and a powerful scripting language • Highly optimized, full ANSI standard, 32-bit clean, FORTRAN 77 compiler • Fully compatible with virtual memory under System 7 • Supports a variety of third-party tools to customize any FORTRAN environment • Power Tools include graphing software, subroutine libraries, and user interface tools • Free technical support.

Our Price **$695.00** (SLSFORT)

## Fortran 77SDK by Absoft

For Power Macintosh includes a globally optimizing native compiler and linker, native Fx™ multi-language debugger, and Apple's MPW development environment.

• The compiler is a full ANSI/ISO Fortran 77 implementation, and includes all MIL-STD 1753 extensions, Cray/Sun-style POINTER, and several Fortran 90 enhancements • MRWE, Absoft's framework library is included in the MIG graphics library • Supports the native Macintosh PPC toolbox • Includes Absoft's Fx debugger which can debug intermixed FORTRAN 77, C, C++, and PPC assembler • The linker compiler, and debugger all run as native PPC tools, and produce Macintosh PPC executables

Our Price **$699.00** (SF77)

## NS BASIC for the Newton

- A fully interactive implementation of the BASIC programming language.
- Runs entirely on the Newton – no host is required.
- Create files, access the built-in soups, and the serial port for input and output.
- Work directly on the Newton, or through a connected Mac/PC and keyboard.

Our Price **$94.99** (SNSBASIC)

## LPA MacProlog

- Comprises a Edinburgh syntax Prolog compiler system set in an attractive multi-window development environment with an integrated program editor, graphical call-graph facilities and an interactive source-level debugger.
- Features high-level access to the Macintosh ToolBox for using graphics, dialogs, windows, icons, resources in a simple and versatile way.
- Includes interfaces to C and Pascal code resources.
- Enables the production of double-clickable distributable applications.
- Optional add-ons tools include flex, Prolog++, MacDBI for Oracle and the MacProlog Dialog Editor.

Programmer Edition **$745.00** (SLPAP) Developer Edition (includes the run-time generator and distribution license) **$1500.00** (SLPAD)

## MacFortran II V 3.3 by Absoft

MacFortran II is a VAX/VMS compatible full ANSI/ISO Fortran 77 compiler including all MIL-STD 1753 extensions

- Comes with the latest version of MPW, and includes SourceBug, and SoftwareFPU
- Includes Absoft's Macintosh runtime Window Environment (MRWE) application framework, and MIG graphics library
- MacFortran II features improved 68040CPU support and is fully compatible with Power Macintosh under emulation

List $595.00 Our Price **$549.00** (SFORT2)

## CodeWarrior Software Development Using PowerPlant by Jan L. Harrington

• C++ programmers will learn to develop object-oriented software applications for the Mac and Power Mac using the PowerPlant environment and the classes that support it. • Covers CodeWarrior 8. • Included CD-ROM contains source code for all the programming examples in the book and Metrowerks CodeWarrior Lite.

List Price: $34.95 Our Price: **$31.45**

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

## Roaster DR2™

**From Natural Intelligence Inc. Get the most out of Sun's new Java™ Programming Language!**

- Write, test and run Java™ applets on the Macintosh in a full-featured Mac development environment
- Features include: project window that includes a finder-like view of packages, Macintosh native compiler, source code editor with powerful search features and intuitive use interface, runtime engine for quick and easy applet testing.
- Requirements : PowerPC, CD-ROM.

List $399.00 Our Price **$299.00** (SROAST)

SEE RELATED CATEGORY: **Dev. Environments**

## TCP/IP Scripting Addition™

SEE RELATED CATEGORY: **Scripting**

- Award-winning AppleScript scripting addition
- Allows you to write scripts using MacTCP™ commands in AppleScript™.
- Send e-mail or files through a script, check if users are logged on (via Finger), automate FTP, Gopher, NetNews, Telnet, and LPR, verify links in HTML documents, and quickly write many other TCP/IP client-server programs.
- Works with AppleScript 1.0 or later and MacTCP 2.0.4 or later. Compatible with Open Transport™ 1.1.

Our Price **$49.00** (STCP)

SEE RELATED CATEGORY: **Tools, Libs & Utilities**

## OOFILE HTML Writer

The first OODBMS framework to offer a complete solution for application authors. Replaceable backend database, currently Faircom's c-tree Plus for cross-platform royalty-free power. PowerPlant and other frameworks integrated with edit fields, database browsers and more. AppMaker users, generate complete applications. User-friendly syntax makes it easier to migrate from FoxPro and the non-OO world. Demo's on CodeWarrior and AppMaker CD's. **$895.00** (SOOFCTREE) for a once-off c-tree bundle, or **$1,095.00** (SOOFSUB) 1year subscription. HTML and character-mode report-writer **$195.00**.(SOOFHTML) Full GUI report-writer, including HTML, is **$495.00** (SOOFGUI). Mac Platform Bundle - includes all Mac frameworks, c-tree, and Report-Writer. **$1,495.00** (SOOFBNDL).

## Providing Internet Services via the Mac OS

**by Carl Steadman and Jason Snell**

- Shows you how to provide Web pages, FTP, Gopher, e-mail, and more with the Mac OS.
- Includes CD-ROM containing all the Mac server software and utilities you need.

List: $34.95

Our Price: **$31.46**

## Learn HTML on the Mac

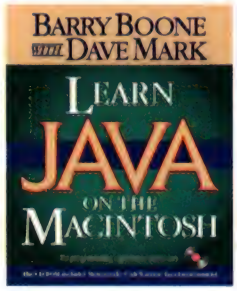
**by Dave Mark and David Lawrence.**

* Shows you how to use HTML 3.0.
* Included CD-ROM contains Macintosh tools to design stunning multimedia Web pages.

List: $29.95 Our Price: **$26.95**

## Learn Java on the Mac

**by Barry Boone with Dave Mark.**




- Easy-to-follow introduction for beginning programmers and Webmasters.
- Takes you through the core concepts of Java.
- Includes: Object-oriented techniques, unique features such as garbage collection, and basic programming concepts such as working with variables, threads and classes.
- Learn to apply this knowledge to write original Java applets for Web pages.

* Includes CD-ROM

List: $34.95 Our Price: **$31.45**

For information: • Voice: 805/494-9797 • Fax: 805/494-9798 • E-mail: marketing@devdepot.com

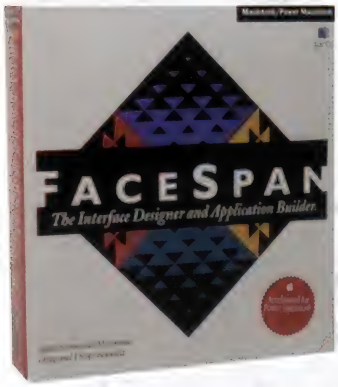
e on these products
http://www.devdepot.com

**Can't find what you're looking for?**
**Check our Web site: http://www.devdepot.com**

## DataScript

• Make your integrated AppleScript solutions database aware-today. • Takes just six lines of AppleScript to make new or existing scripted solutions database aware, and fetch data from RDBM's such as Oracle, Sybase, DB2, or Informix. • Easy to learn, easy to use, and easy to remember. "Inside DataScript" contains lots of easy to follow scripts to reuse in your own solution. • Very attractive licensing schemes.

List $249.00 Our Price **$229.99** (SWDSCRIPT)




## FaceSpan™ v2

**Build applications quickly and easily!**

- Extensible Rapid Application Designer (RAD).
- Combines an interactive, visual interface design environment with the object-oriented power of AppleScript or any OSA language.
- Integrate the capability of scriptable programs into your custom application.
- Display scrolling lists, popup menus, scrolling text, movies, multi-column tables, pictures, icons, buttons, and others.
- Program custom objects using Pascal or C.
- Includes a royalty-free distribution license, for unlimited runtime users, of your FaceSpan-based applications. Plus a FREE UPGRADE to the next version for registered users!

List $199.00 Our Price **$179.00** (SFACESPAN)

## Script Debugger by Late Night Software Ltd.

- A powerful and flexible AppleScript authoring tool – get the most from AppleScript!
- Advanced debugging environment offers single-step script execution with breakpoints.
- Script Debugger dictionary browser features a graphical view of objects provided by scriptable applications.
- Includes Late Night Software Scripting Additions – a collection of more than 70 new AppleScript commands, and Scheduler, a utility that allows you to launch scripts at pre-determined times.

List $129.00 Our Price **$119.00** (SDEBUG)

## PreFab Player by PreFab Software, Inc.

- Lets your scripts query and control otherwise non-scriptable applications, desk accessories and control panels.
- Adds verbs that directly manipulate the Macintosh user interface: choose from menus & pop-ups, select radio buttons, type text, determine the name of the frontmost window, the state of a check box, etc. Balloon help identifies non-standard dialog items.
- Offers a simple, inexpensive way to distribute scripts and updates.
- Use a standard scripting system for automated testing.
- Requires: AppleScript or Frontier

Our Price **$95.00** (SPLAYER)

## ScriptWizard™ 1.5

• Combines the power of a professional development environment with the Mac's ease of use. • Compatible with all Apple® Open Scripting Architecture languages, including AppleScript™. • Includes delivering testing and debugging facilities to improve your productivity. • Emphasizes features that speed script development. • Single-step scripts with true statement-level stepping, watch variable values as scripts execute, jump instantly to frequently used places in a script and find and replace specific text.

List $89.00 Our Price **$84.95** (SWIZ)

## Scripter® by Main Event

- Powerful and easy to use, for both novices and experts.
- Includes Superior vocabulary access – point-and-click assembly of commands and object specifications; command window for experimentation.
- Includes Shortcuts and extended editing capabilities – extensive drag-and-drop, six-function find-and-replace; navigation markers; script library collection facility; many other timesavers for faster scripting.
- Interactive debugging – comprehensive variable watcher, expression evaluation, enhanced trace log, and real single step debugging!

List $199.00 Our Price **$179.00** (SSCRIPTER)

o on these products
http://www.devdepot.com

**Complete info on these products and hundreds more! http://www.devdepot.com**

SEE RELATED CATEGORY:
**Internet Related**

## BBEdit 4.0 from Bare Bones Software

- A powerful, easy-to-learn text editor.
- Adds new features for HTML coders, including a spelling checker and HTML tag palette.
- Accelerated for Power Macintosh; dragging supported everywhere; Internet Config aware; PowerTalk aware.
- Integrated support for Symantec's IDE, Metrowerks CodeWarrior, THINK Reference 2.x, MPW ToolServer, and most other environments.
- Many UNIX style tools, including "grep" searches, file comparisons, and sorting. Multi-file search and replace.
- PopUpFuncs feature lets you jump to a function from a menu.

List $119.00 Our price **$94.00** (SBBEDIT)

## QUED/M 3.0 by Nisus Software

- The programmer's text editor that defined the industry standard for speed and efficiency.
- Power PC native.
- Features integrated support for Symantec C/C++, Metrowerks CodeWarrior 6, and MPW.
- Supports all the major development environments on the Macintosh.
- Dozens of powerful editing features, including unlimited undo and redo, macro language, scripting, text folding, ten editable/appendable clipboards, markers, displaying text as ASCII codes, dynamic coloring of C/C++ keywords/comments, rectangular and non-contiguous selection.
- Includes Celestin Company's APPRENTICE 4.

List $149.00 Our Price **$89.00** (SQUEDM)

## CMaster 2.0 by Jersey Scientific

- Installs into THINK C 5 / 6 / 7 and Symantec C++ for Macintosh and enhances the editor.
- Use its function popup to select a function and CMaster takes you right to it!
- Multiple clipboards and markers, a Function Prototyper, and a GoBack Menu which can take you back to previous editing contexts.
- Almost all features bindable to the keyboard, along over a hundred keyboard-only features like "Add New Automatic Variable." Glossaries, AppleScript and ToolServer support, Macros, and External Tools you create too!

Our Price **$129.95** (SCMASTER)

SEE RELATED PRODUCTS:
**Quick Time Starter Kit**

## Movie Cleaner Pro 1.2 by Terran Interactive

- Compress QuickTime movies
- Powerful and easy-to-use
- Includes: drag and drop batch processing, suspend and resume, adaptive noise reduction, IMA audio compression, high quality crop and resize, A/V fades, gamma correction, de-interlacing and more!
- Automatically suggests the best settings for your application.
- Great for both beginners and experts
- Essential for CD-ROMs or Web sites

Our Price **$189.95** (SMOVIE)

SEE RELATED PRODUCTS:
**Quick Time Official Guide**

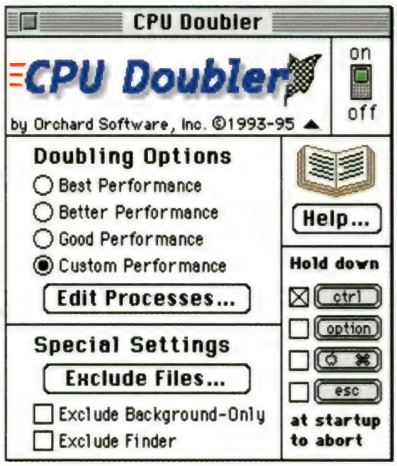
**One & 2-Day SHIPPING** (call for details)

## CodeManager by Metrowerks

SEE RELATED CATEGORY:
Dev. Environments

- Source code control system for the Macintosh.
- Based on and compatible with Microsoft Visual SourceSafe version 4.0.
- Works with any type of binary file including graphic, database, library and executable files.
- Includes support for multi-platform projects, security features, reverse delta versioning of files, comment areas for each revision, project analysis and reporting tools.
- Sold on a subscription basis with two future product updates.
- **Includes a 1 year MacTech subscription.**

Our Price **$399.00** (SCDMGR)




## CPU Doubler by Orchard Software

SEE RELATED PRODUCTS:
Cron Manager

- Performance enhancement utility for the Macintosh.
- Increases the speed of your computer by 100%.
- Works on both the PowerPC and 68K Macintosh.
- Manages computer throughput using a proprietary scheduling algorithm.
- Ensure optimal performance and compatibility.

Our Price **$79.95** (SCPU2X)

## CLImate by Orchard Software

- Command line interface that lets you communicate with your Macintosh using English commands to create, delete, rename, and move files and folders.
- Start applications, format disks, restart your computer, etc.
- Supplements the Finder.
- Includes a BASIC interpreter that lets you script your Macintosh without AppleScript.
- Includes advanced programming constructs: repeat loops, if/then/else conditionals, subroutine calls, etc.
- Implements wildcard characters, enabling you to work on groups of files.
- Comes bundled with sample programs and full documentation.
- Includes a free copy of Cron Manager – chronological event management utility.

Our price **$59.95** (SCLIMATE)

FILE genie PRO

## File Genie Pro by Duet Development

- File Genie Pro lets you search ALL your developer CD-ROMs at once!
- Every Dev CD, OS SDK, CodeWarrior CD, E.T.O., and Bookmark CD you insert is cataloged automatically. A typical Dev CD with 10,000 files is cataloged in under 10 seconds.
- You can also catalog opticals, SyQuests, and floppies.
- File Genie Pro quickly finds any file on any disk by searching hard disks, network servers, and catalogs. Open, show, print, and more. View text and graphics files without opening other applications.
- When acting on an ejected disk file, File Genie Pro tells you which disk to insert, then continues automatically.

Our Price **$89.00** (SGENIE)

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

## Guide Composer™ 1.2 by StepUp Software NEW!

SEE RELATED PRODUCTS: Real World AppleGuide
SEE RELATED PRODUCTS: AppleGuide Complete
SEE RELATED PRODUCTS: Danny Goodman's AppleGuide Starter Kit

- Create powerful Apple Guide help systems for any new or existing Macintosh application.
- Great for commercial developers, shareware developers, in-house developers, and consultants.
- Provides a WYSIWYG development environment: Guide content is developed in Guide windows.
- Design topics, phrases, and panels in the same format as the user will use them. Features are WYSIWYG interface, Topics, phrases, and hierarchical phrases, Coach marks, Fully-Integrated with Apple's Guide Maker (distributed with Guide Composer), compiles scripts automatically, PICTs in Panels, Generated Guide scripts are modifiable.
- Compiled files are 100% AppleGuide-compatible and royalty-free. Easy-to-use.

STEPUP SOFTWARE

New features include:

- Add up to six radio buttons on an Apple Guide panel, each of which can branch to another sequence or topic
- Add up to six checkboxes on an Apple Guide panel, each of which can insert another sequence or topic into the current topic and much more!
- FREE Update to all registered Guide Composer users. Demo is available at http://www.guideworks.com/.

Our Price **$99.00** (SGCOMP)

## ScriptGen Pro by StepUp Software

- Installer script generator which requires no programming or knowledge of Rez.
- Supports StepUp's InstallerPack, StuffIt decompression, Compact Pro decompression, custom packages, splash screens, network installs, and resource installation.

Our price **$169.00** (SSCRPTGEN)

## Step-Up Installer Pack by StepUp Software

- Package of several Installer "atoms" that let developers incorporate graphics, sounds, file compression and custom folder icons into installation scripts.
- Compression formats supported are Compact Pro & Diamond.
- Each atom also available separately.
- Compression requires additional licensing.

Our price **$219.00** (SINSTALL)

## CronManager by Orchard Software

SEE RELATED PRODUCTS: CPU Doubler

- Implements the UNIX Cron facility.
- Open any Macintosh file on a given date and time.
- Simple interface.
- Works with any Macintosh file.
- Cron Manager bundled with CLImate,

Our price **$26.95** (SCRONMGR)

## Tenon Ported Application CD

- Contains hundreds of applications that have been pre-compiled to run on Personal or Professional MachTen 68k systems, such as Unix, Perl, and TCL/TK
- Includes many internet servers such as Gopher, WAIS Ported Application CD Vol. 1 (for 68K Macs).

Our price **$49.95** (SPORTED)

## Rosanne

SEE RELATED CATEGORY: Scripting

- A collection of utilities which offer the user complete control over raw data.
- Sort files, extract selected records, summarize frequency counts, create sample files, perform matching on multiple files, and reformat data to new specifications, all on the desktop, and even on files of a million records or more.
- Supports AppleScript™, enabling the user to link several actions together to complete an entire process.
- Recordable – users may perform a series of actions, and see them translated directly into AppleScript commands.
- Supports multi-tasking and background processing.
- Rosanne Utilities: Copy, Format, Select, Recode, Sort, Match, and Aggregate

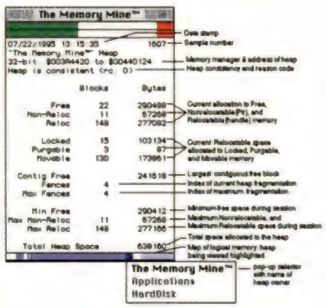
Our Price **$595.00** (SROSANNE)

## Memory Mine by Adianta

SEE RELATED CATEGORY: Dev. Environments

- Monitor heaps, identify problems such as memory leaks, and stress test applications
- Active status of memory in a heap is sampled on the fly: allocation in non-relocatable (Ptr), relocatable (Handle) and free space is shown, as are heap corruption, fragmentation, and more
- Allocate, Purge, Compact, and Zap memory lets users stress test all or part of a program
- Works on Macintoshes with 68020 or later and System 7.0 or later.

List $99.00 Our Price **$94.99** (SMEMMINE)



## QC by Onyx Technology. High performance runtime stress testing for applications.

- Tests include heap checks, purges, scrambles, handle/pointer validation, dispose/release checks, write to zero, de-reference zero as well as other tests like free memory invalidation and block bounds checking.
- Extremely user friendly – ideal for non-programmer testers.
- Also available in Japanese.

List $99.00 Our price **$94.99** (SQC)

SEE RELATED CATEGORY: Dev. Environments

## Last Resort Programmers Edition

- Records every keystroke, command key and mouse event (in local coordinates) to a file on your hard disk
- Great for program testing & debugging, and for technical support and help desks
- Last Resort keystroke files and recovers what you typed – great for data losses caused by power failure, crashes, or accidental deleting.

Our price **$74.95** (BLSTRSRT)

## Spyer by InCider

- Easy to use tool that records all actions (including mouse movement) you perform on a Macintosh computer and then replays them at your preferred speed.
- Recorded data can be saved in files for future use.
- Works as a background process with any Macintosh application and is triggered by user defined Hot Keys.
- Enables the "Continuous Redo" utility and is especially useful for software testing and demonstration.

Our price **$39.00** (SSPY)

## VOODOO

- Version control tool for the simple and clear management of projects in which files are created in numerous versions (variants and revisions).
- Allows both variant and revision control, and it manages not only variants and revisions of single files, but of a whole software project (multi files, multi users, multi variants, access rights, etc.)
- Graphical user interface and is not only suitable for mere source code control but can handle all different kinds of files with amazing compression rates: typical size of delta between arbitrary files 5%
- Please note special prices for multiple copies:

Single license **$229.00** (SVOODO01); 2 pack **$359.00** (SVOODO02); 5 pack **$799.00** (SVOODO05); 10 pack **$1369.00** (SVOODO10); 20 pack **$2399.00** (SVOODO20)

Additional pricing available on request.

## SoftPolish CD-ROM by Bare Bones Software

- The essential tool for software quality assurance on the Macintosh
- Helps you identify inconsistencies with Apple's user interface guidelines, misspelled words, missing resources, and other mistakes
- Provides tools to put the finishing touches on software distribution packages prior to release
- Works independently of any programming language or environment
- Ideal for sanity checking software throughout the development process.

List price: $99 Our price: **$89** (SSOFTPOL)

## LJ Profiler by Lars Jordebo Datakonsult

- Supports profiling of C++ 68K and PowerPC applications compiled with CodeWarrior, CFront or Symantec C++.
- Based on active profiling, i.e. profiling code called at function enter and exit, the browser application lets you follow call chain timings in hierarchical views or separate windows.
- Collect, organize, compare and save profiling data from different versions of your application into a project. Scriptable and recordable with full access to most internal data structures.
- Optional remote profiling and tracking of segment and stack usage. Full source code to what you link into your application.

Our price **$295.00** (SLJPROF)

## PowerTap™

- Accelerates software by tapping into multiple processors.
- Easy API – submit tasks and retrieve results!
- Full error recovery system plus scheduling algorithms for optimal assignments and fastest possible execution.
- Compatible with all Macintosh hardware, software and major compilers.

PowerTap/2, 2 remotes edition List $1,200.00 Our Price **$1,095.00** (SPOWTAP2)
PowerTap/5, 5 remotes edition List $1,900.00 Our Price **$1,795.00** (SPOWTAP5)
PowerTap/n, unlimited edition List $2,700.00 Our Price **$2,495.00** (SPOWTAPN)

**One & 2-Day** SHIPPING (call for details)

## StoneTable

- Replaces all functions found in list manager
- Variable size columns/rows; different font, size, style, forecolor, backcolor per cell; sort, resize, move, copy, hide columns/rows; edit cells/titles in place; titles for columns/rows; multiple lines per cell; grid line pattern/color; greater than 32k data per table; up to 32k text per cell; support for balloon help and binary cell data. Versions for Think C, Think Pascal, MPW C, MPW Pascal, CodeWarrior 6 C.

## StoneTable Extra

- Drag selected cells within table or to other tables.
- Add rows as part of drag; popup menus or check boxes in cells; variable width grid lines; move/drag/resize table in window; clipboard operations on multiple cells.
- Requires StoneTable.

This product can now be ordered in 3 different packages.

### 68K StoneTable

This includes StoneTable, StoneTableExtra for Think Pascal, Think C, MPW C, MPW Pascal, CodeWarrior C, CodeWarrior Pascal

Our price **$175** (SSTABLEX)

### PPC StoneTable

This includes StoneTable PPC, StoneTableExtra PPC for Think C, MPW C, MPW Pascal, CodeWarrior C, CodeWarrior Pascal

Our price **$175** (SSTNXPPC)

### 68K/PPC StoneTable

Includes both packages

Our price **$325** (SSTONEFAT)

## PictureCDEF 1.3 by Paradigm Software

- Professional-level CDEF for creating custom graphical buttons (8-64 pixels) – used in products by Adobe, ProVue, STF Technologies and others!
- Multi-monitor and bit-depth sensitive.
- The button graphic (cicn, ResEdit) can be changed at runtime and even animated with a call-back routine.
- Create distinct buttons in seven variations: MultiState, PushButton, FlexiButton, ToggleButton, ChkButton, PushPictButton and TogglePictButton.
- Manual, sample code and MacApp 3.0 support included.

Full source code: **$95.00** (SPCDEF) Object code: **$45.00** (SPICTOBJ)

## SpellsWell 7 1.0.4

- Award-winning, comprehensive, practical spelling checker that works in batch mode or within applications that incorporate the Apple Events Word Services protocol (e.g., Eudora, WordPerfect, Communicate!, and Fair Witness).
- Checks for spelling errors as well as common typos like capitalization errors, spaces before punctuation, double word errors, abbreviation errors, a/an before vowel/consonant, etc.
- MacTech orders include developer kit with Writeswell Jr., a sample AppleEvents Word Services word-processor and its source code.

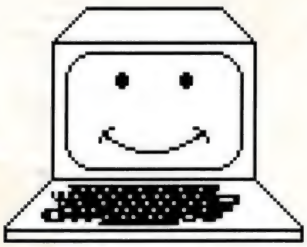
Our price **$74.95** (SSPELL)

## MacWireFrame by Amplified Intelligence

- Create your own virtual reality application with MacWireFrame, a virtual reality application framework.
- Includes a complete library of object oriented graphics routines.
- Easy to understand application frame work, plus an example application program that lets you start solid modeling right away.
- Complete with fully documented source code.
- Guaranteed $49.99 upgrade to the soon to be released, scriptable, MacWireFrame 5.0.

List $299.00 Our Price **$75.00** (SFRAM)

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)



## B-Tree HELPER™ 2.2

- Inexpensive database engine for Macintosh programmers in C source code.
- Uses contiguous fixed length blocks.
- Expands the file as necessary and contracts files when possible.
- Inserts and deletes keys in one or more B-Trees.
- Finds keys equal to, less than, or greater than a given value in a few hundredths of a second.
- Finds lists of records whose keys are equal to, less than, or greater than a given value or are in a range of values.

Our Price **$150.00** (SBTREE)

## Q3S/3dPane/SmartPane

- Full featured 3d graphics library.
- Polyhedra; Gouraud shading; stereoscopic projections; pipeline access; animation and model interaction support; a "triad mouse" to map 2d mouse movement to 3d; and all the regular 3D features.
- 3dPane provides integration with the TCL and provides a view orientation controller. SmartPane provides TCL offscreen image buffering, flicker free animation, and QuickTime movie recording. SmartPane functions in 3d or 2d scenarios.
- All work with C++ compilers or ThinkC 6 and compile to PowerPC or 68K target machines.

Our Price **$192.00** (SQ3)

## dtF

- True relational database system for Apple Macintosh computers.
- Provides a powerful choice for developers who want to create database centered applications with no performance trade-offs.
- Features SQL, full transaction control, error recovery, single user, client server architecture and multi-platform support including DOS, Windows, OS/2 and UNIX.
- The C/C++ API is identical and fully portable across all supported platforms.
- Third-party vendors supporting dtF will be able to offer a variety of advanced features and benefits to their customers royalty free.
- Tools are included for importing, exporting, creating and managing databases and users.
- Supported development environments include: Symantec, MPW, Metrowerks and more. Mac/SDK

List $695.00 Our Price **$679.00** (SDTF)

## AppMaker

- Develop the user interface for a Macintosh application using the original interface builder.
- Just point and click to design your application.
- Creates resources and generates excellent source code.
- Supports most development environments including Metrowerks, Symantec, or MPW;
  C, C++, or Pascal; procedural or object-oriented, using PowerPlant, TCL, or MacApp.
- The generated code uses the Universal Headers to provide PowerMac compatibility.
- Great tool for beginners to learn object-oriented and Macintosh Toolbox programming techniques.
- Includes one-year subscription on CD.

List $299.00 Our Price ~~$279.99~~

SPECIAL PRICE! **$149.00** (SAPPMAKE)

Includes CD-ROM!

## NeoAccess™

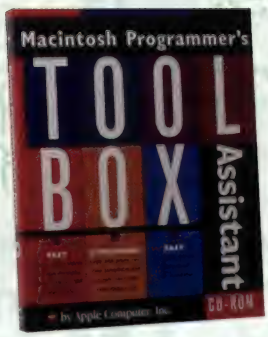
- Full-featured object database engine for use in Macintosh, Windows, Unix and DOS based C++ applications.
- Extended binary trees and binary search algorithms tuned for short access times; dynamically combined, collapsed, and compressed indices; object caching for instant access to previously used objects.
- Build fast, powerful applications in record time!

Our price **$749.00** (SNEO)

Lowest price guaranteed. Feel Good about your purchase! 30-day return policy.

## Inside Macintosh®: Programmer's Toolbox Assistant CD-ROM

**Instant electronic access to Inside Macintosh essentials.**




- Get quick access to reference pages for over 4,000 Toolbox calls in your system software from their development environment.
- Essential information for Macintosh software developers.
- Hypertext links allow programmers to view related topics easily.
- The ultimate electronic reference tool for Macintosh programmers.

List $99.95 Our Price **$49.99** (STBASST)

## Inside Macintosh®: Overview by Apple Computer, Inc.

- An overview of Macintosh programming fundamentals and a road map to the New Inside Macintosh library.
- Covers various programming tools and languages, compatibility guidelines and considerations for worldwide development. 176 pages.
- Great for beginners!

List $22.95 Our Price **$11.50** (BIMOVER)

## Inside Macintosh®: Macintosh Toolbox Essentials

**by Apple Computer, Inc.**

- Covers the heart of the Macintosh. The toolbox enables programmers to create applications consistent with the Macintosh "look and feel".
- Describes Toolbox routines and shows how to implement essential user interface elements, such as menus, windows, scroll bars, icons and dialog boxes. 880 pages.

List $34.95 Our Price **$17.50** (BIMTBOX)

## Inside Macintosh®: More Macintosh Toolbox by Apple Computer, Inc.

- Managers discussed include Help, List, Resource, Scrap and Sound.
- Covers other Macintosh features such as how to support copy and paste, provide Balloon Help, play and record sound and create control panels.

List $34.65 Our Price **$17.33** (BIMMAC)

## Inside Macintosh®: Files by Apple Computer, Inc.

- Describes the parts of the operating system that allow you to manage files.
- Explains how your application can handle the commands typically found in a File menu.
- Provides a reference to the File and Alias Managers, the Disk Initialization and Standard File Packages. 510 pgs.

List $29.95 Our Price **$14.99** (BIMFIL)

## Inside Macintosh®: Files by Apple Computer, Inc.

- Describes the parts of the operating system that allow you to manage files.
- Explains how your application can handle the commands typically found in a File menu.
- Provides a reference to the File and Alias Managers, the Disk Initialization and Standard File Packages. 510 pgs.

List $29.95 Our Price **$14.99** (BIMFIL)

## Inside Macintosh®: Operating System Utilities by Apple Computer, Inc.

- Describes parts of the Macintosh Operating System that allow you to manage various low-level aspects of the operating system.
- Shows how to get information about the operating system, manage operating system queues, handle dates and times, control the settings of the parameter RAM, manipulate the trap dispatch table, and receive and respond to low-level system errors.

List $26.05 Our Price **$13.00** (BIMOPSU)

## Inside Macintosh®: Processes by Apple Computer, Inc.

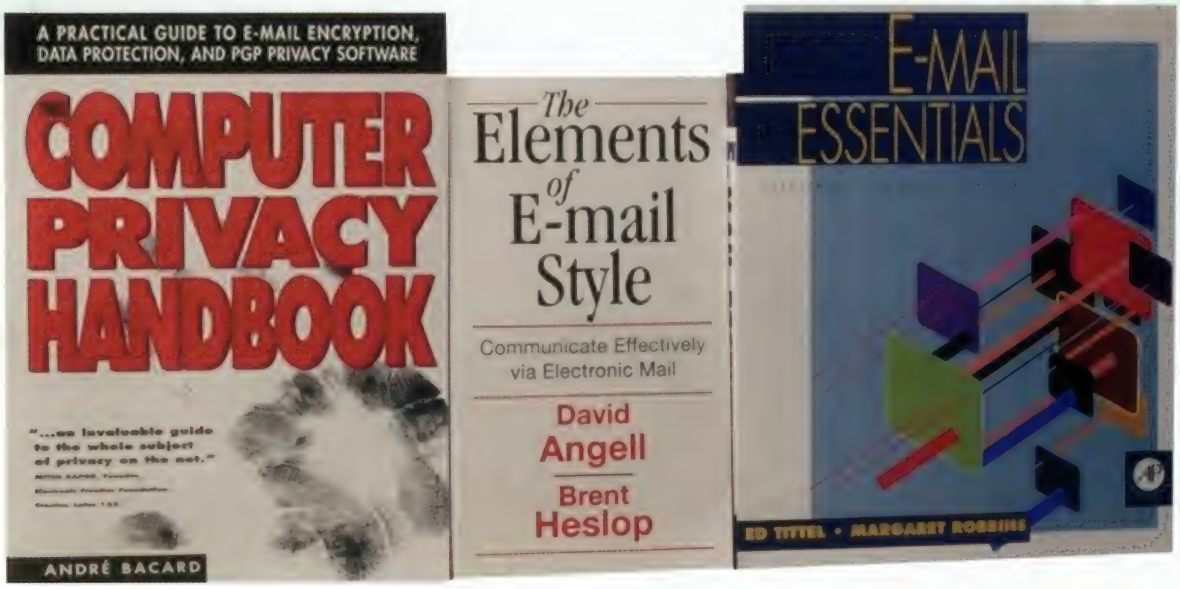
- Describes the parts of the Macintosh operating system that allow you to control the execution of processes and interrupt tasks.
- Shows how you can use the Process Manager to get information about processes loaded in memory.
- A reference for the Vertical Retrace, Time, Notification, Deferred Task, and Shutdown Managers. 208 pages.

List $22.95 Our Price **$11.50** (BIMPROC)

## Inside Macintosh®: Memory by Apple Computer, Inc.

- Describes the parts of the Macintosh operating system that allow you to manage memory.
- Provides detailed strategies for allocating and releasing memory, avoiding low-memory situations, reference to the Memory Manager, the Virtual Memory Manager, and memory-related utilities. 296 pages.

List $24.95 Our Price **$12.50** (BIMMEM)


A PRACTICAL GUIDE TO E-MAIL ENCRYPTION, DATA PROTECTION, AND PGP PRIVACY SOFTWARE
COMPUTER PRIVACY HANDBOOK
ANDRÉ BACARD
The Elements of E-mail Style
Communicate Effectively via Electronic Mail
David Angell
Brent Heslop
E-MAIL ESSENTIALS
ED TITTEL • MARGARET ROBBINS

### The Computer Privacy Handbook

- A practical guide to e-mail encryption, data protection, and PGP privacy software.

List $24.95 Our Price **$22.45** (BPRIV)

### The Elements of E-Mail Style

**by Brent Heslop and David Angell**

- Write solid, effective E-Mail and avoid common pitfalls.

List $14.95 Our Price **$13.45** (BEMAIL)

### E-Mail Essentials

**by Ed Tittel & Margaret Robbins**

- A hands-on guide to the basics of e-mail.

List $24.95 Our Price **$22.45** (BEMAILE)

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

## Cyberpunk Handbook,

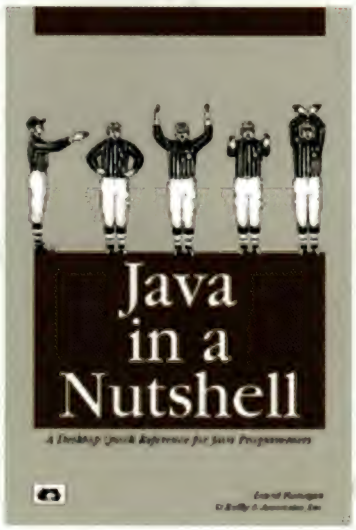
List $9.95 Our Price **$8.95** (BCYBPUNK)

## Planning and Managing Websites

- The definitive guide to setting up and running a Web site on the Macintosh.
- Learn everything you need to know about using WebSTAR, the best known HTTP server software and its shareware predecessor MacHTTP.
- Write CGI applications for your server – in AppleScript and in C.
- CD includes a special version of WebSTAR, plus tons of useful software.

List $39.95 Our Price **35.96** (BPLANWEB)

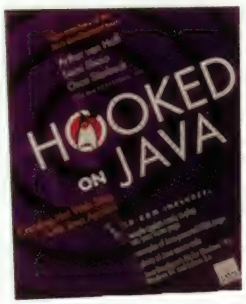
SEE RELATED CATEGORY: Internet Related




## Java in a Nutshell

- A complete quick reference guide to Java, the hot new programming language from SunMicrosystems.
- Contains descriptions of all of the classes in the Java 1.0 API, with a definitive listing of all methods and variables.
- Also contains an accelerated introduction to Java for C and C++ programmers who want to learn the language fast.

List $14.95 Our price **$13.45** (BJAVANUT)




## Hooked on Java

- Written by the Java development team at Sun.
- An introduction to using applets, for Web administrators, designers, and developers.
- Demonstrates how to use applets in your own pages.
- Includes a concise introduction to the Java language, and a CD with tools.

List $29.95 Our Price **$26.95** (BHJAVA)

## Teach Yourself Java for Macintosh in 21 Days

- Add interactivity and multimedia to Web pages!
- Astep-by-step guide to make your Web site come alive.
- Learn the basics of programming Java applets and the concepts behind the Java language.
- Includes CD-ROM with a limited version of Roaster, the first commercial, integrated applet development environment for Java for the Macintosh!

List $40.00 Our price **$36.00** (BJAVAMAC)

## The Instant Internet Guide

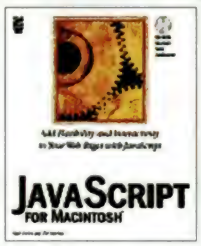
List $14.95 **$13.45** (BINSTANT)

Lowest price guaranteed. Feel Good about your purchase! 30-day return policy.

## JavaScript for the Macintosh

- Allows non-programmers to take advantage of the power of Netscape Navigator.
- Expand the capabilities of your Web page, without having to understand C or C++.
- CD-ROM contains "Wizlets" that allows you to easily create your own JavaScripts
- Takes you step-by-step through programming cross-platform JavaScripts
- Details how to create JavaScripts for JavaScript-aware Web browsers

List $45.00 Our price **$40.50** (BJAVASCRPTJ)



# Developer DEPOT™

## ORDER FORM

http://www.devdepot.com
E-mail: orders@devdepot.com
U.S./Canada: 800-MACDEV-1 (800-622-3381)
Outside U.S./Canada 805-494-9797
Fax: 805-494-9798

**Please fill out as completely as possible to avoid delays in processing your order.**

Name _____

Company _____

Address _____

City _____ State/Province _____

Zip/Postal Code _____ Country _____

Home Phone _____ Business Phone _____

E-Mail Address _____ Fax _____

☐ Cash ☐ Check/Money Order ☐ VISA ☐ MasterCard ☐ American Express Expiration Date _____

Card # _____ Signature _____

**ORDERING INSTRUCTIONS:** Itemize the products you want below. Fill in the Order Total, Sales/Use Tax (for California residents only), and Subtotal. Shipping & Handling charges will be added upon processing. All orders will be shipped via most economical method (e.g., UPS Ground), unless you state otherwise. **Remember to fill out your address and payment information above.**

| PRDUCT NUMBER | ITEM DESCRIPTION | QUANTITY | UNIT PRICE | TOTAL |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

If you are ordering a subscription to **MacTech™ Magazine**, are you a new subscriber, or are you renewing your subscription?

☐ New Subscription
☐ Renewal

*Magazine subscriptions are non-taxable

Order Total _____

Sales/Use Tax (CA residents only)* _____

**SUBTOTAL** _____

FOR OFFICE USE ONLY

Shipping & Handling _____

**TOTAL** _____

Fold Here First

Fold Here (Last) and Tape Closed (not staple)

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS MAIL PERMIT NO. 264 THOUSAND OAKS, CA

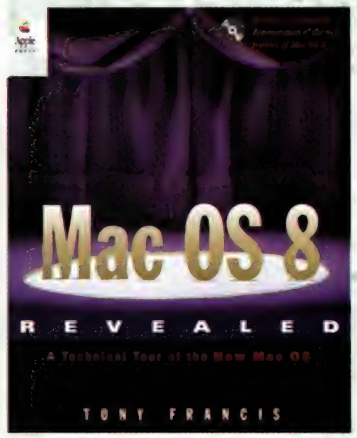
POSTAGE WILL BE PAID BY ADDRESSEE

XPLAIN CORPORATION
P O BOX 5200
WESTLAKE VILLAGE CA 91359-9864

## Mac OS 8 Revealed by Tony Francis

* The first authoritative look at this exciting new operating system.
* A must for Mac developers who want to make their software compatible with Mac OS 8.
* essential for system administrators who plan to upgrade their system.
* Included CD-ROM contains demos of new Mac OS 8 features.

List: $34.95 Our Price: **$31.45**



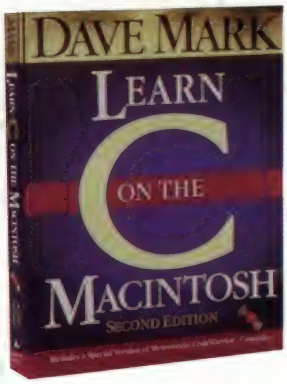

## Designing AppleTalk Network Architectures

**- Network Frontiers Field Manual Series by Dorian J. Cougias, Tom Dell & E.L. Heiberger**

* Learn to update your current network design, or create a new one.
* Thoroughly explains AppleTalk, and provides a complete overview of cabling and hardware issues to maintain a succesful network.
* Includes CD-ROM

List Price: $29.95 Our Price: **$26.95**

One & 2-Day SHIPPING (call for details)




## Learn C on The Macintosh Second Edition By Dave Mark

- New revised edition.
- Easy-to-understand – everything you need to start programming!
- Updated and enhanced exercises that lead you step by step. You'll learn function, variables, pointers datatypes, data structures, file input and output and more!
- Includes CD-ROM with Metrowerks CodeWarrior™ Lite – the hottest Macintosh programming environment (including a PowerPC version).

List $34.95 Our Price **$31.45** (BLEARNC2)

SEE RELATED CATEGORY: **Dev. Environments**

## Discover Programming for Macintosh

- Includes full working version of CodeWarrior along with three online tutorial books and Dave Mark's "Learn C on the Macintosh" converted to AppleGuides.
- Includes C, C++ and Object Pascal compilers for generating 68K Macintosh code, source-level debuggers, object-oriented frameworks (PowerPlant, MacApp), Apple's MPW, complete online documentation and source code examples for all languages and platforms.
- The IDE software has been localized in eight languages plus English. This product is not sold as a subscription.
- **Includes a 3 month subscription to MacTech Magazine**.

Our Price **$79.00**

SEE RELATED CATEGORY: **Dev. Environments**

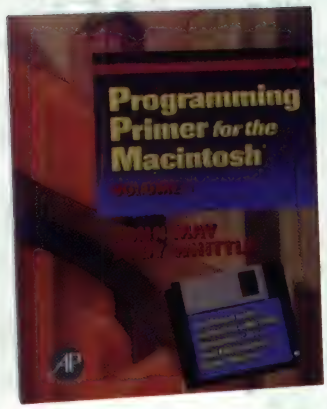
## ScriptBase™ by Main Event Software

- A database for storing persistent objects to be made available for access to AppleScript, Apple's system-level user scripting language for controlling applications on Macintosh® computers.
- Once installed, the database becomes part of the AppleScript system, adding a host of commands to the basic AppleScript vocabulary. Retrieving the objects is simple using AppleScript's natural-language syntax and structure. Objects stored and retrieved in ScriptBase can be accessible any time from any script on the user's computer. These objects can be of any type, including numbers, character strings, lists, records, scripts, and references to disks, files, folders, as well as abstract raw data, to name just a few.
- ScriptBase can be used to maintain system-wide settings, such as sets of preferences, paths to frequently-used files or folders. Complex installations can be made easier by organizing data and scripts within the database's structure.

Our Price **$79.00** (SSCPTBASE)

## Macintosh C Programming Primer Volume I

**Second Edition, Inside the Toolbox Using THINK C by Dave Mark and Cartwright Reed**




- Updated new edition of the Macintosh programming best seller.
- System 7, new versions of THINK C and ResEdit.
- Learn how to use the resources, Macintosh Toolbox and interface to create stand-alone applications.
- 672 pages.

List $26.95 Our Price **$24.25** (BCPRIM1)

## Macintosh C Programming Primer Volume II

**Mastering the Toolbox Using THINK C by Dave Mark.**

- Covers advanced topics such as: Color QuickDraw, THINK Class Library, TextEdit, and the Memory Manager: 528 pgs.

List $26.95 Our Price **$24.25** (BCPRIM2)

## Learn C++ on the Macintosh by Dave Mark.

SEE RELATED CATEGORY:
**Dev. Environments**

- Basic syntax of C++ and object programming.
- Learn how to write, edit, and compile your first C++ programs.
- Features key C++ concepts such as derived classes, operator overloading, iostream functions and more.
- Includes a special version of Symantec C++ for Macintosh. Book/disk package with 3.5" 800K Macintosh disk. 400 pages.

List $36.95 Our Price **$33.26** (BLRNCPP)

## Macintosh Pascal Programming Primer Volume I

**Inside the Toolbox Using THINK Pascal by Dave Mark and Cartwright Reed.**

This tutorial shows programmers new to the Macintosh how to use the Toolbox, resources, and the Macintosh interface to create stand-alone applications with Symantec's THINK Pascal. 544 pages.

List $26.95 Our Price **$24.25** (BPASCPRI)

## Power Macintosh Programming Starter Kit

**by Tom Thompson.**

SEE RELATED CATEGORY:
**Dev. Environments**

- Enter the world of the PowerPC chips.
- Get the scoop on the microprocessors, the RISC architecture, and how to write native code and emulation operations to create software for the Macintosh PowerPC.
- CD-ROM includes a unique compiler for writing code easily.

Includes CD-ROM!

List $39.95 Our Price **$35.10** (BPPCSTART)

## Macintosh Programming Secrets 2nd edition

**By Scott Knaster, and Keith Rollin**

- Macintosh Programming Secrets is divided in two parts.

**Part 1**, "Concepts and Ideas", discusses the evolution of the Macintosh and the standards, customs, and software that shape the system as well as the Macintosh user interface.
**Part 2** "Technical Adventures", presents the skeleton of an application, and then builds upon that framework to describe how to: • Create fancy dialogue boxes • Utilize the new 32 bit QuickDraw developments • Track the mouse with "marching ants" • Manage multiple windows with the Window manager • Copy files within a program • Install the worlds strangest spinning cursor.

List $31.95 Our Price **$28.76** (BPSECRET)

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

## Cyberdog - The Complete Guide to Apple's Internet Productivity Technology

by Jesse Feiler

- Learn all you need about Cyberdog — Apple's new Internet productivity interface.
- Turn everyday documents into glitzy masterpieces of content.
- Provides guidance on extending and customizing Cyberdog by adding or replacing part editors.
- Includes sample Cyberdog solutions.

List Price: $29.95 Our Price: **$26.95**

## Cyberdog Programmers Kit by Apple Computers, inc.

- Apple's official reference.
- A must for developers who want to make customizable Internet access available to all Mac users.
- Included CD-ROM contains all the tools needed to create Cyberdog-aware components.

List: $34.50 Our Price: **$31.05**

## Programming with AppleTalk by Michael Pierce

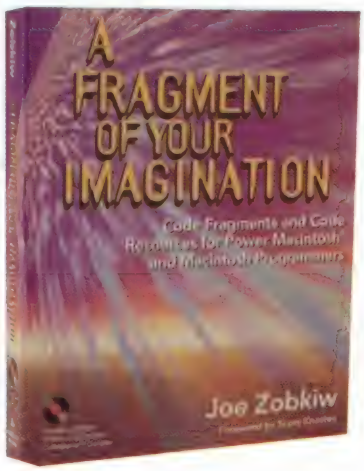
- Programming with AppleTalk is the hands-on guide to understanding and working with AppleTalk. Topics covered include:
- How to create applications and system extensions that run with AppleTalk.
- AppleTalk protocols and the protocol stack, transport media, the Preferred AppleTalk Intefrace, and the storage management.
- Numerous working code examples walk you through using RDEV, INIT, NBP, ATP, and ADSP You will also learn the use of: Synchronous, and asynchronous calls, How to avoid heap fragmentation, And how to configure a Chooser Interface.

List $24.95 Our Price **$22.45** (BPROAT)

## A Fragment of Your Imagination by Joe Zobkiw

- Packed with useful code fragments for the Macintosh and Power Macintosh.
- Hard to find information about techniques used to structure and build fat, safe fat, and accelerated code resources.
- All code is reusable and is provided on the disc, along with Metrowerks Code Warrior Lite. Book/CD-ROM, 528 pages.

List $39.95 Our Price **$35.96** (BFRAG)

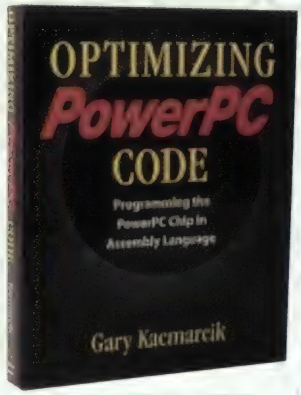


## Programming QuickDraw

- Learn to build color pictures on the Mac
- Learn to maintain the highest degree of compatibility for you applications across the Mac platform
- Learn to perform sophisticated image processing operations with CopyBits( ) and CopyDesk( )
- Learn to enable your applications to take advantage of QuickDraw compliant graphics hardware and accelerators.

List $26.95 Our price **$24.25** (BPROQDRAW)

## Optimizing PowerPC Code: Programming the PowerPC in Assembly Language




- Take full advantage of the potential of the PowerPC by mastering the Assembly Language techniques.
- Learn to produce faster more robust software!

List $39.95 Our Price **$35.96** (BOPTPPC)

## Inside Macintosh®: PowerPC Numerics

**by Apple Computer, Inc.**

- Describes the floating-point numerics environment provided with the first release of PowerPC processor-based Macintosh computers.
- Provides a description of the IEEE standard 754 for binary floating-point arithmetic., and how RISC Numerics compiles with it.
- Shows programmers how to create floating-point values and how to perform operations on floating-point values in high-level languages such as C and in PowerPC assembly language.

List $28.95 Our Price **$26.00** (BIMPPCNUM)

## Inside Macintosh®: PowerPC System Software by Apple Computer, Inc.

- Describes the new process execution environment and system software services provided with the first version of the system software for Macintosh on PowerPC computers.
- Contains information to write applications that can run on the PowerPC.
- Shows how to make your software compatible with the new run-time environment provided on PowerPC-based Macintosh computers. It also provides a complete technical reference for the Mixed Mode Manager, the Code Fragment Manager, and the Exception Manager.

List $24.95 Our Price **$22.45** (BIMPPCSYS)

## AppleScript Finder Guide, English Dialect by Apple Computer, Inc.

- Provides definitions for Finder object classes and commands.
- Write, record, or run scripts that trigger the same desktop actions that you trigger using the keyboard and mouse.

List $19.95 Our Price **$17.95** (BAFG)

SEE RELATED CATEGORY: **Scripting**

## AppleScript Language Guide,

**by Apple Computer, Inc.**

- A complete reference for anyone using AppleScript to modify existing scripts or to write new ones.
- Contains useful information for programmers who are working on scriptable applications or complex scripts.
- Features detailed definitions of AppleScript terminology and syntax in the following categories: Value classes, commands, objects and references to objects, expressions, control statements, handlers, and script objects.
- Includes many sample scripts, discusses advanced topics such as writing command handlers for script applications, the scope of script variables and properties declared at different levels in a script, and inheritance and delegation among script objects.

List $29.95 Our Price **$26.95** (BALG)

SEE RELATED CATEGORY: **Scripting**

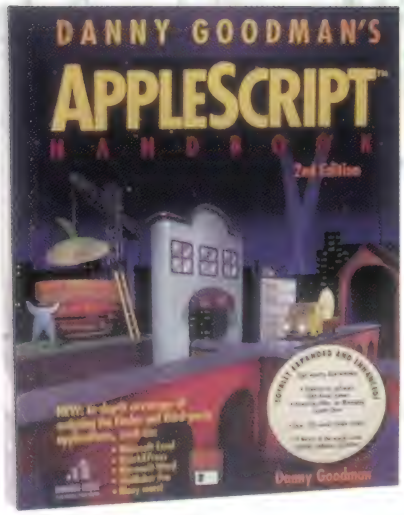
## AppleScript Scripting Additions Guide

**by Apple Computer, Inc.**

- Use the standard scripting additions commands.
- Write scripting additions.

List $18.95 Our Price **$17.05** (BSCRADD)

SEE RELATED CATEGORY: **Scripting**




## Danny Goodman's AppleScript Handbook Second Edition by Danny Goodman

- Customize and extend the capabilities of any Macintosh computer – no programming experience needed!
- Learn to use scripts to enhance the Macintosh environment, automate many processes, link data between applications, and much more.
- All-new examples showing how to integrate AppleScript with the Finder, spreadsheets, desktop publishing programs, graphics applications, databases, telecommunications programs, utilities, and HyperCard.
- Includes 3 1/2" disk with over $100 worth of software, including AppleScript 1.1, valuable utilities, and powerful, ready-to-use scripts.

List $39.00 Our Price **$35.00** (BDGASHB)

Includes Disk!

SEE RELATED CATEGORY: **Scripting**

SEE RELATED CATEGORY: **Tools, Libs & Utilities**

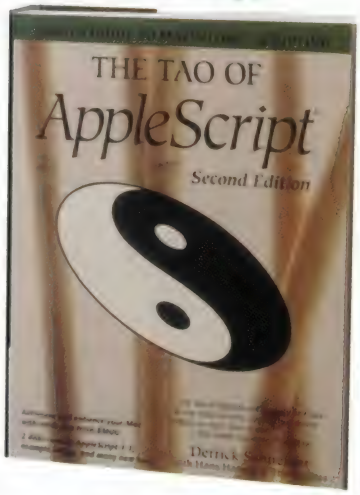
## Applied Mac Scripting

SEE RELATED CATEGORY: **Scripting**

- Learn to design and develop powerful scripts.
- Covers AppleScript™, Frontier, QuicKeys, Tempo II, nShell, FaceSpan Application Builder, Scripting PlainTalk and System 7.5.
- Hands on tutorial shows you how to automate your Macintosh activities by learning how to use the AppleScript and Frontier scripting environments.
- Harness the capabilities of a wide variety of Macintosh applications into the integrated productivity tools. This includes such things as the newspaper script which combines the power of SITcomm, MacWrite Pro, and FileMaker Pro, or QuarkXPress.

List $34.95 Our Price **$31.45** (BAPPLIED)

## The Tao of AppleScript: BMUG's Guide to Macintosh Scripting, Second Edition by Derrick Schneider & Hans Hansen




- A complete, natural introduction to AppleScript programming essentials.
- Learn to customize applications, automate tedious tasks, and create programs without having to use a complex programming language.
- Includes 2 disks containing AppleScript, QuickTime, StuffIt Lite, ResMover, and other helpful utilities.
- Loaded with practical examples for easy learning

List $29.95 Our Price **$26.95** (BTAO)

Includes Disk!

SEE RELATED CATEGORY: **Scripting**

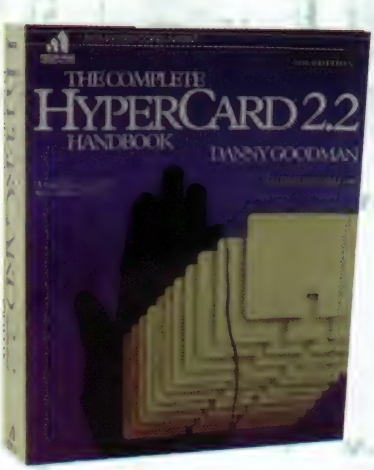
## The Complete AppleScript Handbook by Danny Goodman

Master AppleScript with the definitive book/disk toolkit. This self-contained toolkit teaches you AppleScript from the ground up providing you with tools you need to automate tasks and integrate Macintosh applications. This book explains in detail all commands and usage of the AppleScript language including:

• Issuing command • Describing objects • Working with values, variables, and expressions • Using if-then constructions, loops and subroutines • Error checking and debugging • Scripting Finder-level processes • Using AppleScript with third party applications

List $35.00 Our Price **$31.50** (BAPLSCRHB)

SEE RELATED CATEGORY: **Scripting**




## The Complete HyperCard® 2.2 Handbook Fourth Edition
by Danny Goodman

- The biggest-selling programming Mac book.
- Learn to build working applications using the latest version of HyperCard.
- Covers text, painting tools, extension commands (XCMDs), scripting in HyperTalk, and more.

List $35.00 Our Price **$31.50** (BHYPCRD2)

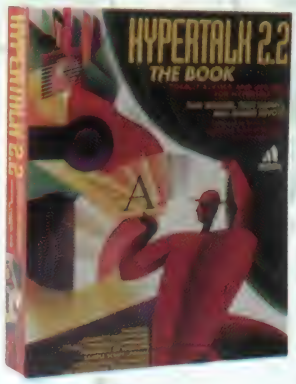
SEE RELATED CATEGORY: **Scripting**

## HyperTalk® 2.2: The Book Second Edition
by Dan Winkler, Scott Kamins, and Jeanne DeVoto

SEE RELATED CATEGORY: **Scripting**

- The most complete, authoritative source on HyperTalk 2.2 programming and troubleshooting.
- Covers each language element of HyperTalk 2.2 (including the odd quirk or bug).

List $35.00 Ours Price **$31.50** (BHYPTAL)




## HyperCard Stack Design by Apple Computer, Inc.

HyperCard Stack Design Guidelines is an essential book for everyone who creates Apple® HyperCard stacks. Included are illustrated discussions of:

- Guidelines for stack development-audience evaluation, subject matter requirements and constraints, mode of presentation, and testing
- Navigation, the importance of making sure users can get around in your stacks
- Graphic Design and illustration- including the use of grids to determine card and background layout
- Text in stacks- placement, readability, and special considerations when writing for the screen
- Music and sound in stacks-as subject matter, reinforcement, entertainment, alert mechanism, or transition

List $21.95 Our Price **$19.95** (BHYPSTA)

SEE RELATED CATEGORY: **Scripting**

## Tog on Software Design by Bruce "Tog" Tognazzini

Respected industry futurist, Tog, presents his vision of our technological future, detailing the steps computer professionals need to take to deliver new technologies that will profit the industry and benefit society in general. Contains Tog's insights on a wide range of topics from quality management to the meaning of standards, and responses to queries supplied by designers and developers.

List $29.95 Our Price **$26.95** (BTOG)

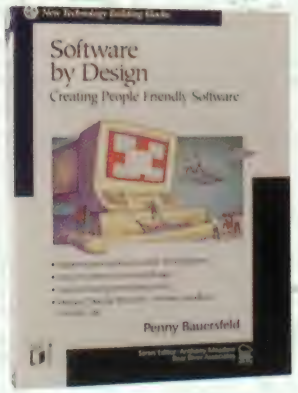
## Object Oriented Program Design by Mark Mullin

- A concise guide to the essential concepts and techniques of OOP design
- Clarifies the key concepts of object oriented programming such as objects, classes, entities, hierarchies, and inheritance
- Uses typical database application to illustrate each OOP topic, to give the programmer a familiar point of reference

List $22.95 Our Price **$20.66** (BOOPRODES)

## Software By Design: Creating User Friendly Software

**by Penny Bauersfeld (Series Editor: Tony Meadow)**




- A thorough how-to for designing software that is easy to learn and comfortable to operate.
- Written from the Macintosh perspective, but compatible with all platforms.
- Stresses user input from initial design, through prototyping, testing and revision.
- Provides tools for analyzing user needs and test responses, plus exercises for sharpening user-oriented design skills.

List $29.95 Our Price **$26.95** (BDESIGN)

## Inside Macintosh®: CD-ROM by Apple Computer, Inc.

- More than 25 volumes in electronic form.
- Includes: QuickDraw™ GX Library, Macintosh Human Interface Guidelines, PowerPC System Software, Macintosh Toolbox Essentials and More Macintosh Toolbox, QuickTime and QuickTime Components.
- Access over 16,000 pages of information with Hypertext linking and extensive cross referencing.

List $99.95 Our Price **$49.99** (BIMCD)

## Inside Macintosh®: Sound by Apple Computer, Inc.

- Describes the parts of the Macintosh system software that allow you to manage sounds.
- Contains information to write applications that can record and play back sounds, compress and expand audio data, convert text to speech, and perform other similar operations.

List $26.95 Our Price **$13.50** (BIMSOUND)

## Inside Macintosh®: Imaging by Apple Computer, Inc.

- Covers QuickDraw and Color QuickDraw.
- Includes general discussions of drawing and working with color.
- Describes the structures that hold images and image information, and the routines that manipulate them.
- Covers the Palette, Color, and Printing Managers, and the Color Picker, Color Matching, and Picture Utilities.

List$26.95 OurPrice **$13.50** (BIMIMAG)

## Inside Macintosh®: QuickTime by Apple Computer, Inc.

- For developers who want to create applications that use QuickTime, the system software that allows the integration of video, animation, and sounds into applications.
- Describes all of the QuickTime Toolbox utilities.
- Provides the information you need to compress and decompress images and image sequences.

List $29.95 Our Price **$14.99** (BIMQT)

## Inside Macintosh®: QuickTime Components by Apple Computer, Inc.

Covers how to use and develop QuickTime components such as image compressors, movie controllers, sequence grabbers, and video digitizers.

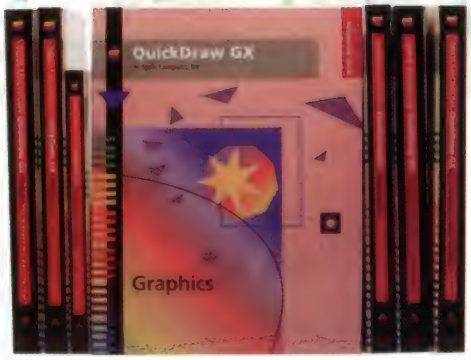
List $34.95 Our Price **$17.50** (BIMQTCOM)

One & 2-Day SHIPPING (call for details)

## Inside Macintosh®: QuickDraw™ GX Programmer's Overview

- Provides an introduction to QuickDraw™ GX, providing an overview of the QuickDraw GX environment from a developer's perspective.
- Introduces the QuickDraw™ GX programming and runtime environments, the relationship between QuickDraw GX and the rest of the Macintosh® systems software and the relationship between QuickDraw GX and Macintosh applications.
- Learn the key elements of QuickDraw GX programming, data structures, object types, and functions used most frequently by QuickDraw GX developers are also covered.
- Provides a series of practical examples demonstrating how to approach programming with QuickDraw GX.

List $24.95 Our Price **$12.50** (BIMGXOV)




## Inside Macintosh®: QuickDraw™ GX Graphics
**by Apple Computer, Inc.**

- Shows how to create and manipulate the fundamental geometric shapes of QuickDraw GX to generate a vast range of graphic entities.
- Demonstrates how to work with bitmaps and pictures, and specialized QuickDraw GX graphic shapes.

List $26.95 Our Price **$13.50** (BIMGXGR)

## Inside Macintosh®: QuickDraw™ GX Objects
**by Apple Computer, Inc.**

Introduces QuickDraw GX and its object structure, and shows programmers how to manipulate objects in all types of programs.

List $26.95 Our Price **$13.50** (BIMGXOBJ)

## Inside Macintosh®: QuickDraw™ GX Printing by Apple Computer, Inc.

- Essential for any developer whose QuickDraw™ GX application supports printing.
- Shows how to support the new printing features of QuickDraw GX, including desktop printers and expandable printing dialog boxes.
- Shows how to use printing-related objects to add custom panels to printing dialog boxes and to create custom page formats.

List $26.95 Our Price **$13.50** (BIMGXPRNT)

## Inside Macintosh®: QuickDraw™ GX Printing Extensions and Drivers by Apple Computer, Inc.

- Essential for developers who want to create extensions to the application printing capabilities of QuickDraw™ GX, or who need to write a printing device driver that works with QuickDraw GX.
- Describes how to create printing extensions and printer drivers, and provides a complete reference to the messages, functions, and resources that they use.

List $29.95 Our Price **$14.99** (BIMGXEXT)

## Inside Macintosh®: QuickDraw™ GX Typography by Apple Computer, Inc.

- Essential for developers who use QuickDraw™ GX to manipulate text.
- Shows how to use QuickDraw GX objects to handle all kinds of text – from plain, unstyled text to complex, mixed-direction and multi-language text with sophisticated stylistic and typographic variations.
- Shows how to create and manipulate the three different types of text shapes supported by QuickDraw GX including text shapes, glyph shapes, and layout shapes.

List $29.95 Our Price **$14.99** (BIMGXTYP)

## Inside Macintosh®: QuickDraw™ GX Environment and Utilities by Apple Computer, Inc.

- Companion to QuickDraw™ GX Objects.
- Contains programming information useful to any developer writing QuickDraw GX applications.
- Describes QuickDraw GX memory management, error handling, debugging, and mathematical functions, as well as conversion from QuickDraw to QuickDraw GX.

List $29.95 Our Price **$14.99** (BIMGXENV)

## 3D Graphics Programming Using QuickDraw 3D
**by Apple Computer, Inc.**

- Incorporate spectacular 3D graphics into your applications.
- Explore QuickDraw 3D, a revolutionary graphics extension to the Mac OS for Power Macintoshes.
- CD contains the complete QuickDraw 3D system itself and a complete database of the QuickDraw 3D API, allowing you instant access to the hundreds of graphics calls via a fast viewing engine. Book/CD-ROM, 640 pages.

List $39.95 Our Price **$35.96** (B3DGRAP)

## 3D Game Machine v1.2 by Virtually Unlimited

- Create lightning-fast 3D arcade games and interactive multimedia applications.
- Ultra-fast rendering – 15 frames per second on a 14" monitor completely texture-mapped, with a PowerMac 6100/60
- Create full "virtual" 3D worlds with six degrees of freedom, free-form texture mapping, shading, material and light properties, convex/cave polygons with unlimited vertices, unlimited light sources, dynamic hidden surface removal, special graphic modes for fast full-screen animation, collision detection, explosion simulation, 3D data importing.
- Simple easy-to-use interface. Runs on all Macs! Works with CodeWarrior.

Our Price **$299.00** + license. (S3DGAME)

## Tricks of The Mac Game Programming Gurus

- For beginning to expert game programmers
- Complete overview of all the necessary components of game programming on the Macintosh.
- Packed with valuable tools, utilities, sample code, CodeWarrior™ Lite and game demos.
- QuickDraw 3D and Power Mac optimization and inside info on how Glypha III was created.
- Hundreds of tried-and-true tricks, tips, and insider secrets from well-known Mac game programming experts

List $50.00 Our Price **$45.00** (BTRICK)

## Black Art of Macintosh Game Programming:

**by; Kevin Tieskoetter.**

- Develop your own 3D games in C on the Mac.
- Includes CD with project files for both Symantec C and Code Warrior
- Create freeform texture-mapped games and polygon graphics
- Control dynamic source code-all compatible as native to the Power Mac
- Write directly to the screen, bypassing QuickDraw

List $39.99 Our price **$35.99** (BBLACK)

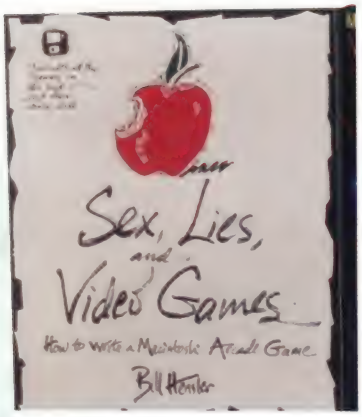
Includes CD-ROM!

## Inside CodeWarrior 9

- Includes CodeWarrior IDE User's Guide.
- This is the printed version of the documentation provided on the CD.
- Covers CodeWarrior, the debugger, and associated tools.

Our Price **$34.95** (BINSCW)

SEE RELATED CATEGORY:
**Dev. Environments**




## Sex, Lies and Video Games by Bill Hensler

Includes CD-ROM!

- A learn-by-example tutorial on the ins and outs of Mac arcade-style game programming in C.
- Features game theory, sprite animation, sound, and interaction techniques.
- A must-read for serious programmer's and hobbyists alike.

List 34.95 Our Price **$31.46** (BSEX)

SEE RELATED CATEGORY:
**Tools, Libs & Utilities**

## Graphic Gems V Edited by Alan W. Paeth

- Loaded with practical tools for implementing new ideas and techniques, to offer working solutions to real programming problems.
- Contains over 40 new gems in ellipses, splines, Bezier curves, and ray tracing – displaying the most recent and innovative techniques in graphics programming.
- Includes a disk with source code from all five volumes. Available in both IBM and Macintosh versions. CONTENTS: Algebra and Arithmetic. Computational Geometry. Modeling and Transformation. Curves and Surfaces. Ray Tracing and Radiosity. Halftoning and Image Processing. Utilities.

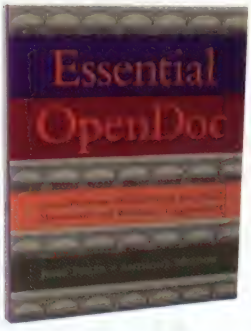
List $49.95 Our Price **$44.95** (BGEMS5)

Includes Disk!

## Advanced Color Imaging on the Mac OS

- Enhance your software's color capabilities with step-by-step instructions.
- Augment the color support supplied with QuickDraw, and QuickDraw GX.
- Use the Pallette Manager to get the best colors on limited displays.
- Match colors between screens and input/output devices (scanners & printers)
- CD includes a complete reference information in both QuickView and Acrobat formats. Plus, a sample application demonstrating ColorSync programming techniques.

List $36.95 Our Price **$33.25** (BADVCI)




## OpenDoc Programmer's Cookbook

- Shows you how to create OpenDoc software components, called parts editors, for the Mac OS Platform.
- Including instructions for setting up the Macintosh Programmers Workshop (MPW) development environment to write OpenDoc software
- Annotated listings of explaining the methods that implement the SamplePart part editor
- Descriptions of other sample part editors created by the OpenDoc engineering team to illustrate more advanced features
- Summary descriptions of software utilities provided with OpenDoc for the Mac OS
- An Introduction to the System Object Model (SOM) technology underlying OpenDoc

List $24.95 Our price **$22.45** (BODCOOK)

## Essential OpenDoc

- Gives an in-depth look at the technical issues of OpenDoc
- Explores the three core technologies that support it's functionality – SOM, OpenDoc's storage mechanism, and the Open Scripting Architecture (OSA).
- Also examines CyberDog, a set of OpenDoc part editors that provides access to Internet services and offers compelling example of the power of OpenDoc development

List $39.95 Our price **$35.95** (BESOD)

## Inside CodeWarrior 9

- Includes CodeWarrior IDE User's Guide.
- This is the printed version of the documentation provided on the CD.
- Covers CodeWarrior, the debugger, and associated tools.

Our Price **$34.95** (BINSCW)

SEE RELATED CATEGORY:
**Dev. Environments**

## ResEdit™ Complete, Second Edition

**by Peter Alley and Carolyn Strange**

- Customize every aspect of your interface form creating screen backgrounds and icons to customizing menus and dialog boxes. 608 pages. Book/disk package.

List $34.95 Our Price **$31.45** (BRESED2)

## Inside PowerPlant Manual

- Create PowerPlant applications using the CodeWarrior IDE and PowerPlant Constructor.
- Full descriptions of major PowerPlant classes and resources.
- Included are the PowerPlant Constructor Manual, including View, TextTraits and Custom Types editing, and PowerPlant Library Reference, covering all classes and functions in PowerPlant.

Our Price **$34.95**

SEE RELATED CATEGORY:
**Dev. Environments**

## OpenDoc Programmer's Guide by Apple Computer, Inc.

- The official reference for the implementation of OpenDoc on the Mac OS.
- Describes the component software revolution and explains how to develop for it on the Mac OS platform.
- Accompanying CD-ROM contains a complete reference to the OpenDoc programming interface, and an extensive collection of tested, reusable sample code.

List $44.95 Our Price **$40.46** (BOPENDOC)

## C++ Programming with CodeWarrior by Jan L. Harrington

- Beginning OOP for the Macintosh and Power Macintosh and Mac OS compatibles.
- Learn object-oriented programming techniques using C++ as the example language and Metrowerks and CodeWarrior as the example compiler.
- Enclosed CD contains example code from the book and a full-function Metrowerks CodeWarrior

List $35.95 Our Price **$32.35** (BCPPCW)

## The ResEdit All Night Diner by David Ciskowski

- An idea-filled menu and introduction to the joys of customizing software.
- Add personality to the Mac by customizing default icons, the text of menus and dialog boxes, cursors, pointers and more.
- Disk features ResEdit, plus lots of sample resources

List $24.95 Our Price **$22.45** (BRESDINE)

Includes Disk!

SEE RELATED CATEGORY: **Dev. Environments**

## C++ Programming W/MacApp by David Wilson, Larry Rosenstein & Dan Shafer

- Learn the secrets to unlocking the power of MacApp®, Apple's development environment for C++
- Learn to design complex windows and views using the ViewEdit tool
- Learn to support multipage text and graphics with only five lines of code
- Learn to support Undo for menu commands and drawing operations that use the mouse.

List $34.95 Our Price **$31.46** (BCPPMACAP)

SEE RELATED CATEGORY: **Tools, Libs & Utilities**

## Programming in Symantec C++ for the Macintosh by Judy May and John Whittle

- An introduction to object-oriented programming, the C++ language, and Symantec C++ for the Macintosh.
- Great for both programmers and beginners alike.
- Covers everything from the basics to advanced features of Symantec C++.
- Includes helpful examples of C++ code that illustrate object-oriented programs.

List $29.95 Our Price **$26.95** (BPSYMCPP)

SEE RELATED CATEGORY: **Dev. Environments**

## Symantec C++ Programming by Neil Rhodes & Julie McKeehan

Symantec C++ Programming for the Macintosh is a tutorial for getting up and running in the Symantec C++ environment, while mastering the techniques of object-oriented programming.

- Explore the Symantec C++ environment, from debugging a program and using resource utilities to building applications and creating objects
- Design programs for compatibility with multiple application frameworks
- Learn how to use the new Visual Architect, Inspector, and templates.

List $45.00 Our Price **$40.50** (BSYMCPP)

SEE RELATED CATEGORY: **Dev. Environments**

## Metrowerks CodeWarrior Programming by Dan Parks Sydow

- Includes CodeWarrior Lite, and Full Coverage of PowerPlant™.
- The best information on Metrowerks CodeWarrior, giving full coverage to the Gold Edition.
- CD includes Code Warrior Lite.

List $39.95 Our Price **$35.95** (BCWPROG)

Includes CD-ROM!

SEE RELATED CATEGORY: **Dev. Environments**

## Dan Shafer Presents the Power of Prograph CPX

- Master the revolutionary graphical object-oriented programming language.
- Step by step course through three interrelated projects of increasing complexity.
- Learn Prograph language, CPX classes and object editors.
- Includes disk with all code in the book.

List $49.95 Our Price **$19.95** (BDANPRO)

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

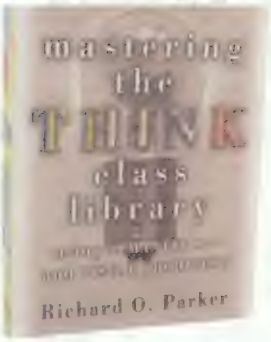
## Visual Programming with Prograph CPX

by Scott B. Steinman and Kevin G. Carver

- An introduction to the language and a guide for advanced users, for both Macintosh and Windows-based machines.

List $34.00 Our Price **$30.60** (BVISPRO)

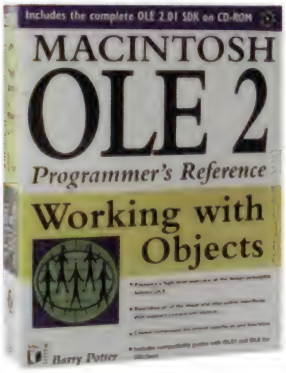
## Mastering the THINK Class Library by Richard Parker




- Provides a thorough examination of Symantec's extensive Class Library and the Visual Architect.
- A complete description of the structure and operation of the TCL includes explanations of all code generated by the Visual Architect, any necessary custom code, and the operation of this code.
- Visual Architect tutorials provide you with a step-by-step approach for simplifying the development of complex Macintosh applications. 496 pages.

List $29.95 Our Price **$26.95** (BMASTERTCL)

SEE RELATED CATEGORY: Dev. Environments




## Macintosh OLE2 Programmer's Reference: Working with Objects

- Complete reference to the extensible protocol of Object Linking and Embedding, version 2.01 for Macintosh System 7.
- Describes the visual and interactice interfaces that support the component objects.
- Provides details of the OLE 2.01 for the Macintosh user Interface, addresses the issues of object class registration, shows how to implement the drag and drop objects from one application to another, covers the interface that exposes the basic embedding functionality, includes descriptions of API functions and more!

Includes CD-ROM!

List $44.95 Our Price **$40.45** (BOLE2)

## Danny Goodman's Apple Guide Starter Kit

**by Danny Goodman and Jeremy Joan Hewes.**

- Create your own Apple Guide databases quickly and easily, without having to learn a scripting language, write coded files, or use several different files and programs
- Includes advice and tips on how to design a good Guide, from planning and creation through testing, revising, and indexing. Book/disk, 320 pages.

List $34.95 Our Price **$31.46** (BDGAGSK)

## MacsBug Reference & Debugging Guide For MacsBug version 6.2

**by Apple Computer, Inc.**

- MacsBug is an assembly-language-level debugging tool
- Macros, templates, dcmds, and other resources for making debugging easier
- Macintosh memory management and the operating system as they relate to low level debugging
- How to display and set memory and process registers
- Disassemble memory, and set execution breakpoints
- Discipline, a tool for testing the validity of toolbox parameters
- Debugging strategies you can use to find and cure common bugs
- Includes MacsBug 6.2.

List $34.95 Our Price **$31.46** (BBUGREF)

## AppleGuide Complete

**by Apple Computer, Inc.**

Includes CD-ROM!

- Covers Guide Maker, the software you use to build and test guide files.
- Learn about the complete cycle of designing as well as advanced topics such as scripting and coding guide files. Book/CD-ROM, 544 pages.

List $39.95 Our Price **$35.96** (BAPLGD)

SEE RELATED CATEGORY: Tools, Libs & Utilities

## Programming For The Newton: Software Development using NewtonScript

**by Julie McKeehan and Neil Rhodes. Foreword by Walter R. Smith**

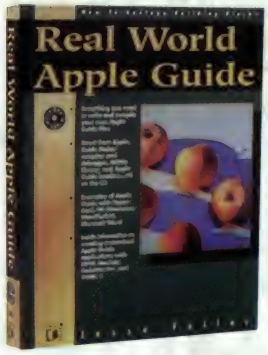
- An indispensable tool for Newton programmers.
- Includes disk with sample Newton application from the books, as well as demonstration version of Newton Toolkit (NTK) – the complete development environment for the Newton®.
- A Publication of AP Professional May 1994, Paperback, 393 pp.

List $29.95 Our Price **$26.95** (BPROGNEWT)

SEE RELATED CATEGORY: Dev. Environments

Lowest price guaranteed. Feel Good about your purchase! 30-day return policy.

## Real World AppleGuide For The Mac

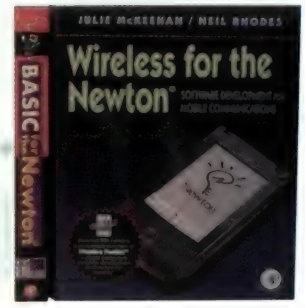



- An introduction to AppleGuide for programmers. It explains its design and function, plus how to design your own guides using AppleScript.
- Includes a disk of sample AppleGuides for AppleGuide-compliant applications.

Includes Disk!

List $39.95 Our Price **$35.95** (BREALWLD)

SEE RELATED CATEGORY: Tools, Libs & Utilities




## Wireless For The Newton: Software Development for Mobile Communications

**by Julie McKeehan and Neil Rhodes**

- Learn to develop Newton® software on the Macintosh.
- Hands-on Newton environment training with sample code
- Includes disk with sample source code for a Newton application, as well as demonstration NTK™ – the complete development environment for the Newton®.

List $34.95 Our Price **$31.45** (BWIRELESS)

## Programming for the Newton Using NS BASIC

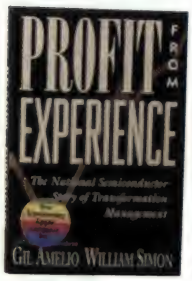
**by John Schettino & Liz O'Hara**

Includes Disk!

- Program on Macintosh, Windows-based PC, or on the Newton itself.
- Straight-forward "programming by example" approach – you'll be writing Newton programs right away.
- Includes 3.5" disk containing Demonstration NS BASIC and over fifty example programs. (Newton not included)

List $35.95 Our Price **$32.35** (BPROGNEWT)

SEE RELATED CATEGORY: Dev. Environments



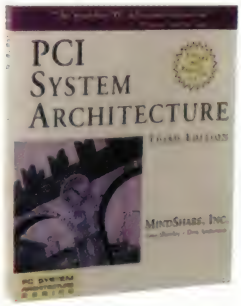

## Profit From Experience by Gil Amelio and William Simon

**Written by the new CEO of Apple**

- The story of the transformation of National Semiconductor – how Amelio and his management team took it from it's worst loss in 30 years to the highest earnings in it's history • Includes: 6 core business issues and why they are critical to success • The TEAM program • 10 personal attributes to strive for • Amelio's guidelines for General Managers • Attributes to look for when hiring.

List $24.99 Our price **$22.45** (BPROFIT)

## PCI System Architecture, Third Edition by MindShare




Describing revision 2.1 of the Peripheral Component Interconnect (PCI) bus specification, this book explores PCI's relationship to the rest of the system. It includes an in-depth treatment of PCI to PCI bridges, the PCI BIOS, the 66MHz PCI bus, and more. 592 pages. List $34.95 Our Price **$31.46** (BPCISYS)

## PowerPC System Architecture

**by MindShare Inc. and Tom Shanley**

- Describes the hardware architecture of the PowerPC systems, providing clear, concise explanation of the PowerPC specification, the template upon which all PowerPC processors are designed.
- Includes the specs for both the 32 and 64 bit implementations including • supervisor privilege level facilities • logical memory addresses • I/O and memory mapped I/O • address translation for segments, pages, and blocks • virtual paging • interrupts.

List $34.95 Our Price **$31.46** (BPPCARCH)

## Inside Macintosh®: AOCE Application Interfaces

**by Apple Computer, Inc.**

- Shows how your application can take advantage of the system software features provided by PowerTalk system software and the PowerShare collaboration servers.
- Add electronic mail capabilities to your application, write a messaging application or agent, store information in and retrieve information from PowerShare and other AOCE catalogs.
- Add catalog-browsing and find-in-catalog capabilities to your application, write templates that extend the Finder's ability to display information in PowerShare and other AOCE catalogs.
- Add digital signatures to files or to any portion of a document, and establish an authenticated messaging connection.

List $40.45 Our Price **$20.25** (BIMAOCE)

## Inside Macintosh®: Interapplication Communication

**by Apple Computer, Inc.**

Shows how applications can work together. How your application can share data, request information or services, allow the user to automate tasks, communicate with remote databases.

List $34.95 Our Price **$17.50** (BIMIAPP)

## Inside the Macintosh Communications Toolbox

**by Apple Computer, Inc.**

- The definitive reference to the Macintosh Communications Toolbox, an integral part of the System 7 Macintosh Toolbox that enables developers to create communications applications or add communications features to other applications.
- Describes all of the routines that provide programmers with standard access to important communications services and in addition enables programmers to extend the reach of the Macintosh into non-Apple environments.

List $24.95 Our Price **$12.50** (BCOMM)

## Inside Macintosh®: Networking

**by Apple Computer, Inc.**

- Describes how to write software that uses AppleTalk networking protocols.
- Describes the components and organization of AppleTalk and how to select an AppleTalk protocol.
- Provides the complete application interfaces to all AppleTalk protocols, including ATP (AppleTalk Transaction Protocol), DDP (Datagram Delivery Protocol), and ADSP (AppleTalk Data Stream Protocol), among others.

List $29.95 Our Price **$14.99** (BIMNET)

## Inside Macintosh®: AOCE Service Access Modules by Apple Computer, Inc.

- Describes how to write a software module that gives users and PowerTalk-enabled applications access to a new or existing mail and messaging service or catalog service.
- Shows how to write a catalog service access module (CSAM), a messaging service access module (MSAM), and AOCE templates that allow a user to set up a CSAM or MSAM and add addresses to mail and messages.

List $26.95 Our Price **$13.50** (BIMAOCES)

## Inside Macintosh®: Text by Apple Computer, Inc.

- Describes how to perform text handling, from simple character display to multi-language processing.
- Covers Font, Script, Text Services, and Dictionary Managers, in addition to QuickDraw Text, TextEdit, and International and Keyboard Resources.

List $39.95 Our Price **$19.99** (BIMTEXT)

## Inside Macintosh®: Devices by Apple Computer, Inc.

- Describes how to write software that interacts with built-in and peripheral hardware devices.
- Learn how to write and install your own device drivers, desk accessories, and Chooser extensions.
- Communicate with device drivers using the Device Manager; access expansion cards using the Slot Manager; control SCSI devices using SCSI Manager 4.3 or the original SCSI Manager.
- Communicate directly with Apple Desktop Bus devices; interact with the Power Manager in battery-powered Macintosh computers, and communicate with serial devices using the Serial Driver.

List $29.95 Our Price **$14.99** (BIMDEV)

## Inside Macintosh®: X-Ref by Apple Computer, Inc.

Is a fast access to all the information in Inside Macintosh. Inside Macintosh X Ref; Provides programmers with a quick and easy way to find the exact information they need in this definitive suite of books, (all 26 volumes). It is indexed by topic, volume, chapter, and accompanying page number.

List $19.95 Our Price **$9.99** (BIMXREF)

**Order Toll-free**
**800-MACDEV-1**
(800-622-3381)

All entries in this index are alphabetized.
For your convenience the product names are **bold**, book names are *italicized* and company names are in plain text.


*3D Game Machine* ..... 25
*3D Graphics Programming Using QuickDraw 3D* ..... 24
Absoft ..... 5
Adianta ..... 10
*Advanced Color Imaging on the Mac OS* ..... 26
Amplified Intelligence ..... 12
*AppleGuide Complete* ..... 28
*AppleScript Finder Guide* ..... 20
*AppleScript Language Guide* ..... 20
*AppleScript Scripting Additions Guide* ..... 20
*Applied Mac Scripting* ..... 21
**AppMaker** ..... 14
**B-Tree Helper** ..... 13
Bare Bones Software ..... 8
**BASIC for the Newton** ..... 5
**BBEdit** ..... 8
Bowers Software ..... 14
*Black Art of Macintosh Game Programming* ..... 25
*C++ Programming w/MacApp* ..... 27
*C++ Programming with CodeWarrior* ..... 26
**CLImate** ..... 9
**CMaster** ..... 8
**CodeManager** ..... 9
**CodeWarrior** ..... 3
*Complete AppleScript Handbook* ..... 21
*Complete HyperCard 2.2 Handbook* ..... 22
*Computer Privacy Handbook* ..... 16
**CPU Doubler** ..... 9
**CronManager** ..... 10
*Cyberpunk Handbook, The Real Cyberpunk Fakebook* ..... 16
*Dan Shafer Presents the Power of Prograph CPX* ..... 27
*Danny Goodman's AppleGuide Starter Kit* ..... 28
*Danny Goodman's AppleScript Handbook* ..... 21
**DataScript** ..... 7
*Discover Programming for Macintosh* ..... 17
**dtF** ..... 3
Duet Development ..... 9
*E-Mail Essentials* ..... 16
*Elements of E-Mail Style* ..... 16
Emerson Kennedy ..... 11
*Essential OpenDoc* ..... 26
Excel Software ..... 14
**FaceSpan** ..... 7
**File Genie Pro** ..... 9
Fortner Research ..... 4, 5
**Fortran 77SDK** ..... 5
*Fragment of Your Imagination* ..... 19
**FrameWorks Magazine/Disks** ..... 2
*Graphic Gems V* ..... 25
**Guide Composer** ..... 10
*Hooked on Java* ..... 16
*HyperCard Stack Design* ..... 22
*HyperTalk 2.2: The Book* ..... 22
InCider ..... 11
*Inside CodeWarrior 8* ..... 26
*Inside Macintosh: AOCE Application Interfaces* ..... 30
*Inside Macintosh: AOCE Service Access Modules* ..... 30
*Inside Macintosh: Devices* ..... 30
*Inside Macintosh: Files* ..... 15
*Inside Macintosh: Imaging* ..... 23
*Inside Macintosh: Interapplication Communication* ..... 30
*Inside Macintosh: Macintosh Toolbox Essentials* ..... 15
*Inside Macintosh: Memory* ..... 15
*Inside Macintosh: More Macintosh Toolbox* ..... 15
*Inside Macintosh: Networking* ..... 30
*Inside Macintosh: Operating System Utilities* ..... 15
*Inside Macintosh: Overview* ..... 15
*Inside Macintosh: PowerPC Numerics* ..... 20
*Inside Macintosh: PowerPC System Software* ..... 20
*Inside Macintosh: Processes* ..... 15
*Inside Macintosh: QuickDraw GX Environment and Utilities* ..... 24
*Inside Macintosh: QuickDraw GX Graphics* ..... 24
*Inside Macintosh: QuickDraw GX Objects* ..... 24
*Inside Macintosh: QuickDraw GX Printing* ..... 24
*Inside Macintosh: QuickDraw GX Printing Extensions and Drivers* ..... 24
*Inside Macintosh: QuickDraw GX Programmer's Overview* ..... 23
*Inside Macintosh: QuickDraw GX Typography* ..... 24
*Inside Macintosh: QuickTime* ..... 23
*Inside Macintosh: QuickTime Components* ..... 23
*Inside Macintosh: Text* ..... 30
*Inside Macintosh: X-Ref* ..... 30
*Inside PowerPlant Manual* ..... 26
*Inside the Macintosh Communications Toolbox* ..... 30
*Instant Internet Guide* ..... 16
*Java in a Nutshell* ..... 16
*JavaScript for the Macintosh* ..... 16
Jersey Scientific ..... 8
**Last Resort Programmers Edition** ..... 11
Late Night Software ..... 7
*Learn C on the Macintosh* ..... 17
*Learn C++ on the Macintosh* ..... 18



Developer DEPOT™

**Can't find what you're looking for?**
**Check our Web site: http://www.devdepot.com**

**LJ Profiler**....11
**LPA MacProlog**....5
**LS Fortran**....5
**LS Object Pascal CD-ROM**....4
**MacFortran II**....5
**MachTen Power Unix**....4
*Macintosh C Programming Primer Volume 1*....18
*Macintosh C Programming Primer Volume 2*....18
*Macintosh OLE2 Programmer's Reference: Working with Objects*....28
*Macintosh Pascal Programming Primer Volume I*....18
*Macintosh Programming Secrets*....18
*MacsBug Reference & Debugging Guide*....28
**MacTech CD-ROM**....2
**MacTech Magazine**....2
**MacTech Mouse Pad**....2
**MacTutor, Best of**....2
**MacWireFrame**....12
**MADACON '93 CD-ROM**....2
*Mastering the THINK Class Library*....28
**Memory Mine**....10
Metrowerks....3, 9
*Metrowerks CodeWarrior Programming*....27
**Mjølner BETA System**....5
**Movie Cleaner Pro**....8
Natural Intelligence, Inc.....6
**NeoAccess**....13
NeoLogic....13
Nisus Software....8
*Object Oriented Program Design*....22
Onyx Technology....11
**OOFile**....13
**OOFILE HTML Writer**....6
*OpenDoc Programmer's Cookbook*....26
*OpenDoc Programmer's Guide*....26
*Optimizing PowerPC Code: Programming the PowerPC in Assembly Language*....20
Orchard Software....9, 10
Paradigm Software....12
*PCI System Architecture*....29
**Personal MacTen**....4
**Picture CDEF**....12
*Planning and Managing Websites*....16
*Power Macintosh Programming Starter Kit*....18
*PowerPC System Architecture*....29
**PowerTap**....11
**PreFab Player**....7
PreFab Software, Inc....7
**Presenting Magic Cap**....3
*Profit from Experience*....29
**Programmer's Toolbox Assistant CD-ROM**....15
*Programming for the Newton using NS BASIC*....29
*Programming for the Newton: Software Development using NewtonScript*....28
*Programming in Symantec C++ for the Macintosh*....27
*Programming QuickDraw*....19
*Programming with AppleTalk*....19
**Q3S/3dPane/SmartPane**....13
**QC**....11
Quasar Knowledge Systems....3
**QUED/M**....8
*Real World AppleGuide for the Mac*....29
*ResEdit All Night Diner*....27
*ResEdit Complete*....26
**Roaster**....6
**Rosanne**....10
**Script Debugger**....7
**ScriptBase**....21
**Scripter**....7
**ScriptGen Pro**....10
**ScriptWizard**....7
*Sex, Lies and Video Games*....25
**SmalltalkAgents**....3
**SoftPolish CD-ROM**....11
*Software by Design: Creating User Friendly Software*....23
**SpellsWell**....12
**Spyer**....11
**Step-Up Installer Pack**....10
StepUp Software....10
Stone Tablet....12
**StoneTable**....12
**Symantec C++ for 68k**....4
**Symantec C++ for Power Macintosh**....4
*Symantec C++ Programming*....27
Symantec Corporation....4
*Tao of AppleScript: BMUG's Guide to Macintosh Scripting*....21
**TCP/IP Scripting Addition**....6
*Teach Yourself Java for Macintosh in 21 Days*....16
Tenon Intersystems....4, 10
**Tenon Ported Application CD**....10
Terran Interactive....8
**THINK Pascal**....4
*Tog on Software Design*....22
*Tricks of the Mac Game Programming Gurus*....25
*Visual Programming with Prograph CPX*....27
Vivistar Consulting....13
**VOODOO**....11
*Wireless for the Newton: Software Development for Mobile Communications*....29
Xplain Corporation....2, 5

INDEX

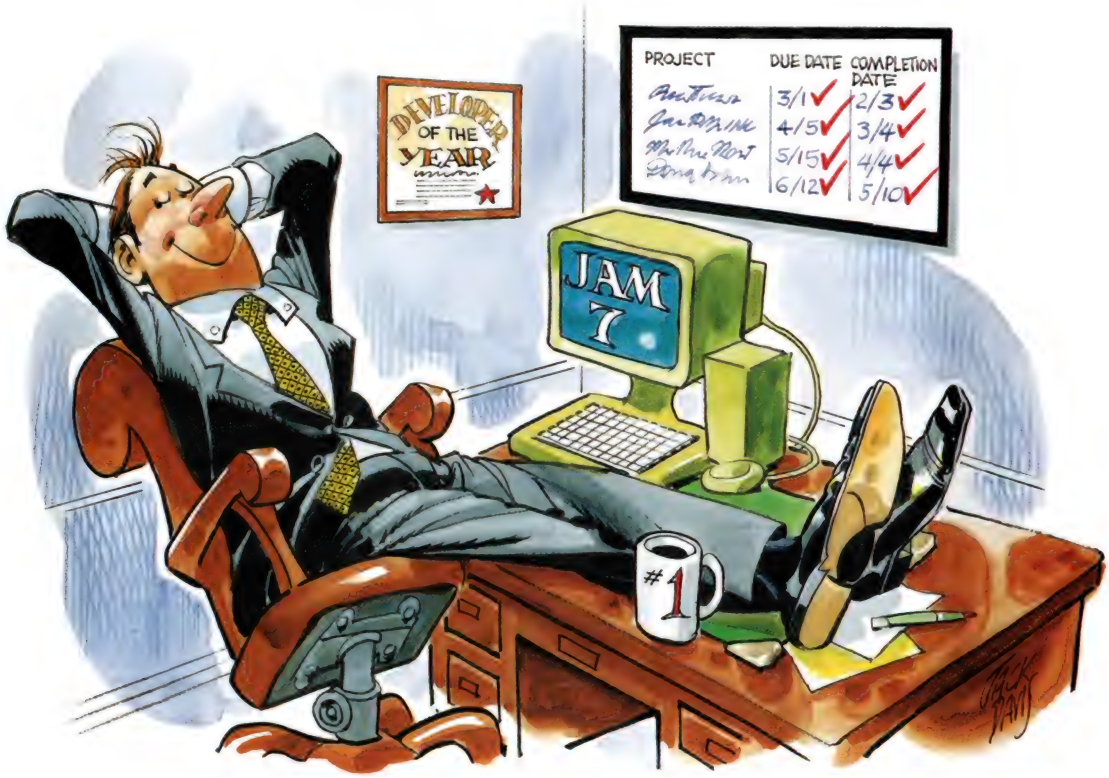
Developer DEPOT™

**Complete info on these products and hundreds more! http://www.devdepot.com**



1-800-622-3381 • Outside U.S. 805-494-9797 • Fax: 805-494-9798

# Life is Sweet with JAM!




## Get JAM and enjoy success in building and deploying client/server and Web applications

In a world where requirements change and deadlines don't, only JAM gives you what you need to build powerful, completely portable client/server, 3-tier and Web applications. JAM's cross-platform RAD capabilities enable you to successfully develop and deploy demanding applications across any platform, database, GUI or Web Browser.

Are you willing to base your project's success on a tool that only *promises* portability? JAM lets you build great looking applications that are fully portable *today*. If you need to build serious client/server, 3-tier or Web applications, call JYACC for a free demonstration kit or white paper.

- ✓ Desktop ease with the power of a 2nd-generation tool
- ✓ Unrivaled portability between Windows® 3.1, NT, 95; Macintosh®; OS/2 Warp®; Character, Motif and Web Browsers
- ✓ Unparalleled database access to Oracle®, Sybase®, Informix®, ODBC and more
- ✓ Automatic SQL generation with full transaction control
- ✓ Visual repository-driven development of both client and application server
- ✓ Centralized control over application objects via inheritance
- ✓ Unique architecture for team development
- ✓ No runtimes

**Find out for yourself how sweet life can be with JAM!**

**Call 1 800 458-3313.**

Or E-mail: sweetlife3@jyacc.com for a free demonstration kit or white paper.
Visit our Web site at http://www.jyacc.com
For international inquiries call: 1-212-267-7722 or FAX 1-212-608-6753

JYACC
Meeting the needs of professional developers for over 18 years

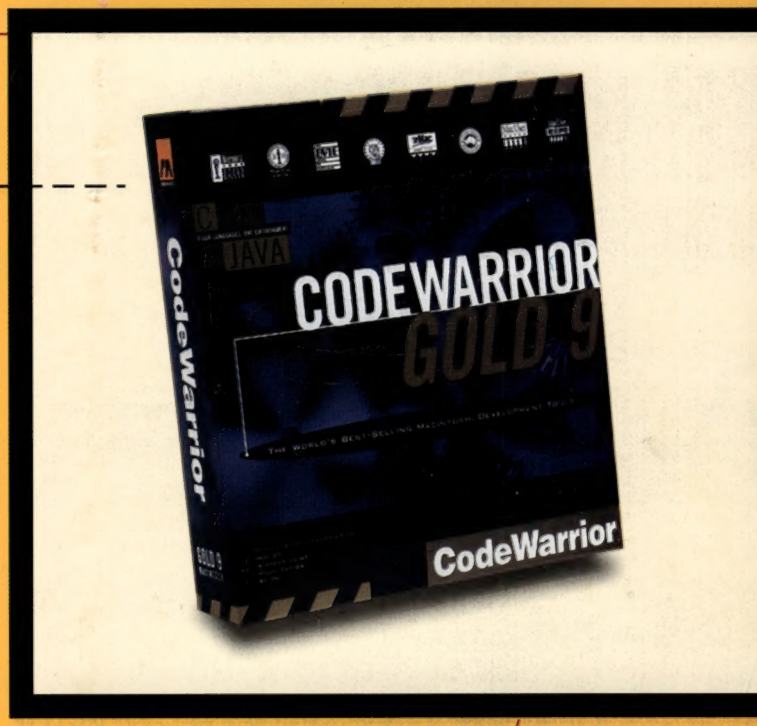
BRAZIL (55) 11 816 6228 • FINLAND (358) 3486 4000 • FRANCE (33) 1 46 92 45 44 • GERMANY (49) 40 79 70 07 0 • HRVATSKA (385) 1 242 116 • ISRAEL (972) 3 557 3675
ITALY (39) 2 45 2701 • MEXICO (52) 5 663 0405 • RUSSIA (7) 095 288 1924 • SAUDI ARABIA (966) 2 665 8406 • SINGAPORE (65) 220 8322 • SLOVENIA (386) 61 1405 004
SPAIN (34) 1 804 0625 • SWEDEN (46) 8 15 10 10 • SWITZERLAND (41) 21 991 9041 • THAILAND (66) 2 513 3559 • THE NETHERLANDS (31) 70 320 9214 • UNITED KINGDOM (44) 171 814 6660

JAM is a registered trademark of JYACC, Inc. Other trademarks are the property of their respective owners.

# The future is definitely not running away from you.

## Java™ tools for CodeWarrior 9 are here.

CodeWarrior, the award-winning Mac-hosted development suite now includes full-featured Java™ tools! CodeWarrior 9 introduces the Java plugin compiler, linker, preference plugins, source-level debugging, and Java class browsing! All this and more, completely integrated into a Java development environment that you could expect only from CodeWarrior.




# CodeWarrior®

### CodeWarrior Gold 9 - $399

The choice for multiple OS development from the Mac™ OS. Includes C / C++, Object Pascal and **new Java compiler plugins** for multiple targets. Compiles for 68K Macintosh®, Power Macintosh™, Windows® 95 and Windows NT™ for x86, Magic Cap™and the Be™ OS.

### Discover Programming™ with Java - $99

Learn how to program using Java. Designed for both novice and experienced programmers, Discover Programming with Java includes the most complete toolset available on the Macintosh, an electronic version of *Learn Java on the Macintosh* by Barry Boone with Dave Mark and 6 Apple® Guide files, the electronic tutorials that take you step-by-step through the learning process.

Macworld 1996 World Class Award Winner · 1994 Software Product of the Year · BYTE · MacLife · MacMac · MacUser · MacUser

For More Information Call 1-800-377-5416

System requirements: Macintosh 68020, 68030 or 68040, or Power Macintosh 601, 603 or 604 processor. Minimum of 8-16MB RAM, CD-ROM drive to install software. ©1996 Metrowerks Corporation. All rights reserved. All products are trademarks of their respective companies.

metrowerks®

# MACTECH MAGAZINE BRINGS YOU TIPS, TRICKS AND SOLUTIONS THAT KEEP YOU CURRENT AND UNRUFFLED.

## AT 33% OFF THE COVER PRICE!

It's not *just* a magazine!
You can also get:

- CD-ROMs
- Books
- Disks
- Free on-line support

If you're a Macintosh programmer, this magazine has spoken *your language* since 1984.

*For Macintosh Programmers & Developers*

MacTech™

MAGAZINE

**To place your order, tear off, complete and return this order form.**

First Name _____ Last Name _____

Company _____ Title _____

Address _____ City _____

State/Province _____ Zip/Postal Code _____ Country _____

Phone No. _____ Fax No. _____ Electronic Address (i.e., Internet, AppleLink, etc.) _____

Check here if: ☐ Renewal ☐ New Subscription How many people read your copy of MacTech Magazine? _____

Payment Method: ☐ VISA ☐ MasterCard ☐ American Express ☐ Check/Money Order Enclosed (US Funds Only)

Credit Card Number _____ Signature _____ Expiration Date _____

☐ US Magazine Subscription: $47 for 12 issues

☐ Canadian Magazine Subscription: US$59 for 12 issues

☐ International Magazine Subscription: US$97 for 12 issues

☐ Domestic Magazine and Disk Subscription: $124 for 12 issues

☐ Canadian Magazine and Disk Subscription: US$156 for 12 issues

☐ International Magazine and Disk Subscription: US$194 for 12 issues

☐ MacTech CD-ROM. Contains every issue since 1984. See the MacTech Mail Order listing for product, pricing and shipping information.

☐ Product from Mail Order Dept (see back of magazine and fill in the products you'd like to order: _____

_____

_____

**California residents add 8.25% sales tax on all non-subscription orders.**

ORDER ENV 10/94

MacTech Magazine, formerly MacTutor, is *the* Macintosh Programming Journal written especially for programmers like yourself. Each month, MacTech Magazine brings you:

- Insider secrets on Macintosh programming
- The latest shortcuts – from massive speed enhancements to faster coding methods
- Full source code listings available on disk
- Hot new product reviews important to your Macintosh development world
- Coverage of a broad range of languages and environments
- Articles for those new to programming on the Macintosh
- Up-to-date developer news, products, and *more!*

## INFORMATION – THE NAME OF THE GAME.

Nothing changes faster than Macintosh computer technology. In each issue, we provide you with the information *you* need to know – directly from the experts in the field. Always on the cutting edge, MacTech Magazine is an invaluable source to overcome programming problems and the day-to-day issues that plague Macintosh programmers.

## AN ONGOING RESOURCE – NOT JUST A MAGAZINE. DISKS & CD-ROMS TOO!

In addition to the monthly publication, an annual compilation of each year of MacTech Magazine is available in book form. Each year, MacTech Magazine also produces a CD-ROM containing all the code and articles for every MacTech Magazine published since 1984. Finally, full source code listings are available on disk that can be ordered either as a subscription or by individual issue. Additionally, these files are available free to subscribers on the major online services.

## A LOT OF MAGAZINES SAY THEY PAY FOR THEMSELVES – THIS ONE REALLY DOES!

Again and again. Every time a MacTech Magazine article solves just *one problem* for you, MacTech Magazine has paid for itself in time savings alone! In short, MacTech Magazine will help you to be a better informed, much more productive and happier Macintosh developer. To maximize your programming efforts, complete and return the above order form.

**NOW, SAVE 33% OFF THE COVER PRICE!**

*Don't miss even one issue!*

*For Macintosh Programmers & Developers*

MacTech™ MAGAZINE

Xplain Corporation, P.O. Box 5200, Westlake Village, CA 91359-5200
Voice: 805/494-9797 • Fax: 805/494-9798
Internet: custservice@xplain.com • CompuServe: 71333,1063
AppleLink: MT.CUSTSVC

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST-CLASS MAIL PERMIT NO. 264 THOUSAND OAKS, CA

POSTAGE WILL BE PAID BY ADDRESSEE

XPLAIN CORPORATION
P O BOX 5200
WESTLAKE VILLAGE CA 91359-9864